

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-75279

(43) 公開日 平成10年(1998) 3月17日

(51) IntCl. ⁸	識別記号	序内整理番号	F I	技術表示箇所
H 0 4 L 29/10			H 0 4 L 13/00	3 0 9 A
H 0 4 M 11/00	3 0 2		H 0 4 M 11/00	3 0 2

審査請求 未請求 請求項の数 2 書面 (全 63 頁)

(21) 出願番号 特願平9-155695

(22) 出願日 平成9年(1997) 5月9日

(31) 優先権主張番号 6 4 5 0 2 0

(32) 優先日 1996年5月9日

(33) 優先権主張国 米国 (U S)

(31) 優先権主張番号 6 6 7 2 6 7

(32) 優先日 1996年6月20日

(33) 優先権主張国 米国 (U S)

(71) 出願人 590000879
テキサス インストルメンツ インコーポ
レイテッド
アメリカ合衆国テキサス州ダラス、ノース
セントラルエクスプレスウェイ 13500

(72) 発明者 ウィリアム シー、ティム
アメリカ合衆国テキサス州マッキニー、ノ
ースビュー ドライブ 2720

(72) 発明者 ウォルター ワイ、チェン
アメリカ合衆国テキサス州プラノ、ランコ
ム ドライブ 8001

(74) 代理人 弁理士 浅村 皓 (外3名)

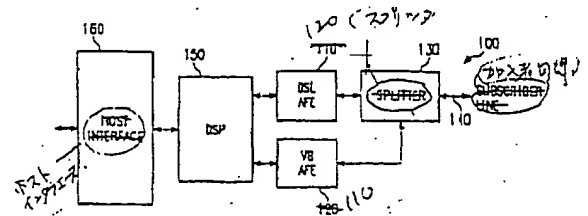
最終頁に続く

(54) 【発明の名称】 デジタル加入者回線モデム信号を復調する方法及びモデム

(57) 【要約】

【課題】 デジタル加入者回線 (DSL) モデムにおいて、高いビット伝送速度を要するデータに対しても、モデムを複雑かつ高価にすることのない、受信DSLモデム信号の復調の方法、及びこのためのモデムを提供する。

【解決手段】 本発明のモデムは、音声帯域周波数帯域及びこれより高い周波数帯域において選択的に動作し、多重回線コードを支援する。このモデムは、デジタル信号処理装置 (DSP) を使用し、これにより離散マルチトーン (DMT) 及び搬送波無しAM/PMが同じハードウェア・プラットフォーム上で実施できる。モデムは、回線状態及びサービス対費用要求を満足させるため、実時間で、所望の回線伝送速度について交渉する。回線コード及び回線伝送速度についての交渉は、2つのモデム間で、トーンを交換することにより各通信セッションの初めに実施する。



【特許請求の範囲】

【請求項1】 受信したデジタル加入者回線モデム信号を復調する方法であつて、

上記受信した信号をバンドパスフィルタを通過させるステップと、

上記信号を復調するため、上記受信した信号を第1の高い周波数帯域からダウンサンプリングして第2の低い周波数帯域における上記信号の両像を作るステップとからなる、受信したデジタル加入者回線モデム信号を復調する方法。

【請求項2】 モデムであつて、少なくとも1つの段階における可変的にスケールしたデータを用いた多段階の計算による変換を採用するデジタル信号処理装置からなるモデム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、デジタル加入者回線モデム信号を復調する方法及びモデムに関し、特に、マルチモード・デジタル・モデム並びにマルチモード・デジタル・モデムのための方法及びソフトウェアを採用したシステムに関する。

【0002】

【従来の技術】 従来の音声帯域モデムは、公衆交換電話回線網(RSTN)を通じて計算機ユーザをエンド・ツー・エンドに接続できる。しかし、音声帯域モデムの伝送スループットは、RSTNのインタフェース点でのバンドパスフィルタ及びコードにより強制される3.5kHz帯域幅に起因して約40kbpsより低く制限される。他方、計算機ユーザの撚線対の電話加入者回線は、より広い使用可能帯域幅をもっている。加入者回線の長さに依存して、50dBの損失での帯域幅は1MHzと広くできる。ローカル加入者回線に基づく伝送システムは、一般的にデジタル加入者線(DSL)と呼ばれる。

【0003】 娯楽(例えば、ビデオ・オン・デマンド)及びデジタル型式の情報(インターネット)への対話型電子アクセスに対する消費者の要求が増加しており、この要求は従来の音声帯域の能力を実際上超過している。これに対して、各家庭への光ファイバリンク、直接衛星送信、及び広帯域同軸ケーブルの様な種々の方法が提案されている。しかし、これらの方法は、しばしば余りに高価であり、そこで、現存する家庭への同軸ケーブルを使用するケーブル・モデム及び種々の、家庭と電話会社中央局(CO)を接続している現存する銅電線の撚線対を使用するの高ビット伝送速度デジタル加入者線(DSL)モデムの様なより安価な代替が現れている。

【0004】 幾つかのデジタル加入者線(DSL)技術が、異なる応用に対して開発されている。元の2B1Qデジタル加入者線技術は、ISDN基本速度アクセ

スチャネルU-インタフェースとして使用されている。高ビット伝送速度デジタル加入者線(HDSL)技術は中継器なしT1サービスとして使用されている。

【0005】 DSL技術の使用の従来技術の一例は、電話回線のための非対称デジタル加入者線(ADSL)シグナリングで、これは標準団体により通信システム仕様書として定義されており、これは住宅からCO(上流)への低速データストリームと、COから住宅(下流)への高速データストリームを提供する、非対称デジタル加入者線(ADSL)は、従来の音声電話通信、例えば、普通の古い電話サービス(POTS)に影響なしに動作を準備している。ADSLの上流チャンネルは簡単な制御機能又は低速データ転送を提供するだけである。高速の下流チャンネルは、より高いスループットを与える。この非対称の流れは、ビデオ・オン・デマンド(VOD)の様な応用には好ましい。

【0006】 非対称デジタル加入者線(ADSL)モデムは、典型的には、対で設置され、1つのモデムは家庭に、他方は、その家庭にサービスしている電話会社中央局(CO)に置かれる。ADSLモデムの対は、同じ撚線対の両端に接続され、各モデムは撚線対の他方のモデムと通信できるだけであり、中央局は、そのADSLモデムから提供されるサービス(例えば、映画、インターネット等)へ直接接続される。図6は、中央局及び1個が消費者の家庭、パーソナルコンピュータ又はTVセット・トップ・ボックスに置かれるADSLモデム(図6はモデムのために「ADSL」よりは「DSL」を使用する)を発見的に示す。ADSLモデムは、音声帯域周波数より高い周波数で動作するので、ADSLモデムは音声帯域モデム又は電話会話と同時に動作しても良い。

【0007】 典型的なADSL使用システムは、中央局(CO)に置かれ映画又は他のデータ強化内容を提供できるサーバ、及び住宅に置かれデータを受けて再組み立てすると共に制御情報をCOへ送り返すセット・トップ・ボックスを含む。有意義な表示又は下流内容の使用にはモデムを通して一様なデータ速度が要求される。この一様なデータ速度の要求に起因してADSLシステムは、ある動作条件及びある速度だけの下で機能するように主として設計される。もし加入者線が品質要件を満たせば、ADSLモデムは機能でき、さもなければ新しい回線機器を設置するか、又は回線品質を改善しなければならない。

【0008】 特に、ANSI標準ADSLは、現存の撚線対で家庭(下流)へ每秒6百万ビット(Mbps)まで伝送し、また家庭(上流)から每秒640キロビット(Kbps)まで受信することを要求する。

【0009】 ADSLモデムは、電話システムを通じてデジタル通信のために現在使用されている音声帯域モデムとは、幾つかの点で異なる。家庭における音声帯域

モデムは、本質的にデジタルビットを音声帯域における変調されたトーンに変換し、従って、これらの信号は、あたかも電話機の中で発生した通常の音声信号であるかの様に伝送されることができる。受信している家庭のモデムは、そこで受信信号からデジタルビットを再生する。現在のITUのVシリーズ音声帯域モデムの標準（例えば、V.32及びV.34）は、33.6Kbpsまでのビット伝送速度での伝送を要求し、たとえ、これらの速度は、実時間ビデオに対するものより遥かに遅く、またインターネット図形に対して遅すぎるものである。これに対して、ADSLモデムは、音声帯域より高い周波数範囲で動作し、これは高いデータ伝送速度を許している。しかし、撚線対加入者線は、周波数及び線長と共に増加する歪みと損失をもち、従って、ADSL標準データ伝送速度は、加入者線の長さに対して最大達成可能速度により決められ、例えば、26ゲージ線に対して3000m（3km）、又は24ゲージ線に対して4kmである。

【0010】音声帯域モデムデータ伝送速度は、少なくとも次の要因により制限される。1）中央局における回線カードのサンプリング速度は、僅かに8kHzである。2）回線カード上に使用されるA/D又はD/A変換器の低いビット解像度は、動的範囲を減少させる。

3）加入者線（撚線対）の長さ及び何等かの関連した電氣的損傷。ADSLモデムは、最初の2つの要因を回避できるが、加入者線長及び電氣的不具合の影響を受ける。図16は、2つの現存する線の寸法に対して、加入者線の容量がどのように減少するかを示している。長さに対する同様な容量の減少はどのような撚線対加入者線にも適用される。

【0011】図14は、単純なADSLモデムをブロック型式で示し、その送信ハードウェア30は、ビットエンコーダ36、逆高速フーリエ変換38、P/S40、デジタルアナログ変換器42、伝送のためのフィルタ及びドライバ44、及びトランス46を含む。受信部分32は、トランス及びフィルタ48、アナログデジタル変換器50、回線歪み補償のための等化器52、S/P54、高速フーリエ変換56、及びビットデコーダ58を含む。送信部分から受信部分へのエコー打ち消し回路は信号漏洩を抑圧するために含ませてもよい。ADSL標準は、256個の4kHz搬送波帯域に分割されたDMTスペクトルをもつ離散マルチトーン（DMT）を使用し、また可変数のビットを他の搬送波帯域に関係なく各搬送波帯域にロードするため配列（コンステレーション）の直角振幅変調（QAM）型を使用している。

【0012】搬送波当りのビットの数は訓練期間の間に決定され、この期間にテスト信号が加入者線を通じて受信モデムへ送信される。受信信号の測定された信号対雑音比に基づき、受信モデムは、最適のビット割当てを決定し、より多くのビットをより強壮な搬送波帯域上に置

き、そしてこの情報を送信モデムへ戻す。

【0013】コード化されたビットの変調は、512点（ポイント）逆高速フーリエ変換を用いて極めて効率良く行なわれ周波数領域でコード化されたビットを時間領域信号に変換し、これを2.048MHz（4×512）サンプリング速度を用いてD/A変換器により撚線対上に置く。受信ADSLモデムは、この信号をサンプリし、高線フーリエ変換でコード化ビットを再生する。

【0014】離散マルチトーン（DMT）が、ADSL標準に対する回線コードとして選択済みである。典型的なDMTシステムは、送信機側FET及び受信機側方向FETを利用する。理想的には、チャネル周波数歪みは受信機FETに続く周波数領域等化器により補正できる。しかし、受信機FETブロックの初めにおけるチャネルの遅延拡大は、前のブロックからの記号間干渉を含んでいる。この干渉は、現在のデータのブロックと無関係であるので、周波数領域等化器により丁度打ち消すことができる。典型的な解決は、プリフィックスのデータのブロックを、FETデータのブロックがD/Aに送られる前に送信機側のFETデータのブロックの前に加えることである。プリフィックス（prefix）データは、FETデータのブロックの最後のセクションの反復コピーである。

【0015】受信機側では、受信信号はウィンドウ処理され、周期的なプリフィックスのデータを除去する。もし、チャネルのインパルス応答の長さがプリフィックスの長さより短かければ、前のFETデータのブロックからの記号間干渉は完全に除去される。周波数領域等化器技術は、そこでDMTサブチャネルの間の記号間の干渉を除去するために適用される。しかし、チャネルのインパルス応答は、その時々により変化するので、チャネルのインパルス応答の長さがプリフィックスの長さより短かいという保証はない。適合時間領域等化器がチャネルのインパルス応答の長さをプリフィックスの長さ以内に短くするため典型的に要求される。

【0016】時間領域等化器訓練手順は以前から検討されている。多重搬送波変調システムのための等化器訓練手順システム、ジェイ・ビー・チョウ、ジェイ・エム・シオフィ及びジェイ・エイ・シー・ビンガム、1993、通信に関する国際会議、pp. 761-765、ジュネーブ、（1993、5月）及びこれに対応する訓練シーケンスはDMTを持つADSL標準及び時間領域等化器（TQE）のための勧告された訓練シーケンス、ジェイ・ビー・チョウ、ジェイ・エム・シオフィ及びジェイ・エイ・シー・ビンガム、AN E1.4委員会寄稿No. 93-086。

【0017】次の特許はDMTモデムに関する。多重搬送波チャネルにおけるビット割当てに関する米国特許第5,400,322号明細書、帯域幅最適化に関する米国特許第5,479,447号明細書、エコー打ち消しに

関する米国特許第5,314,596号明細書、等化器に関する米国特許第5,285,474号明細書。

【0018】代替のDSLモデムの提案は、QAM、PAM、及び搬送波無しAM/PM(CAP)のようなDMT以外の回線コードを使用する。勿論、ISDNは、より多くのデータチャネルを提供するため160KHz以上の搬送波の2ビット-1四元(2B1Q)4レベル記号振幅変調を使用する。

【0019】CAP回線コードは、典型的には、同位相及び直角多重レベル信号を使用し、これらの信号は、直角通過帯域フィルタによりフィルタされ、そして次に伝送のためアナログに変換される。図15は、CAP回線コードを使用するDSLモデムの送信機321及び受信機325のブロック図を示し、両者は等化器750及びエコー打消し(キャンセル)327を含む。

【0020】次の特許は、CAP回線コードに関する。多次元通過帯域伝送に関する米国特許第4,994,492号明細書、エコー打消しに関する米国特許第4,682,358号明細書、及び等化器に関する米国特許第5,052,000号明細書。

【0021】CAP又はDMT、又は他の回線コードを使用するモデムは、本質的に3つのハードウェアのセクションを持つ。(i)加入者回線上のアナログ信号をデジタル信号に変換し、そして加入者回線上へ伝送のためデジタル信号をアナログ信号に変換するアナログフロントエンド、(ii)デジタル信号を情報ビットストリームに変換し、そして随意に誤り訂正、エコー打ち消し、及び回線等化を与えるデジタル信号処理回路、及び(iii)情報ビットストリームとその源/到着地の間のホストインタフェース。

【0022】

【発明が解決しようとする課題】しかし、これらのDSLモデムには次の問題点がある。1)ビデオのために高いビット伝送速度となり、これによりモデムが複雑となり、かつ高価である。2)モデムのビット伝送速度は、ある一定距離に対しては最適化されるが、それより短い加入者回線に対しては効率が悪く、またそれより長い加入者回線には使用できない。3)DMT又はCAPのどちらかは、与えられた異なる条件(例えば、雑音等)に対して良く動作し、これらの条件は、DSLモデムが接続される特定の加入者回線には存在することもあり、存在しないこともある。本発明は、従来技術におけるこれら及び他の欠点を克服する。

【0023】

【課題を解決するための手段】本発明は、7000m(21,000ft)までの長さの標準電話撚線対回線に使用される新しい高速モデムを提供する。この新しいモデムは、MDSL、中間帯域デジタル加入者線(mid-band digital subscriber line)と呼ぶ。本発明のMDSLモデムは、上

流及び下流に伝送される信号を分離するため周波数分割多重を使用する。MDSLに対する変調方式は任意であるが、採用してもよい2つの特定の変調方式は、QAM/CAP及び離散マルチトーン(Discrete Multitone:DMT)である。中央局(CO)にある本発明のMDSLモデムと遠隔ユーザ(RU)端にあるMDSLモデムとの間の同期を達成するための立ち上げ手順は、本発明の一部として提供される。

【0024】MDSLの1つの実施のために選ばれる特定の変調方式の1つは、搬送波無しAN/PM(CAP)である。CAPは同期のために別個のトーンを使用しない。同期は、送信されたデータ信号を直接使用して達成される。立ち上げの時、実際のデータが送信される前に、CAP受信機内の等化器を訓練するため特別のデータシーケンスが使用される。

【0025】本発明は、予め選択された共通の回路を使用して、音声帯域及び高音声帯域(音声帯域より高い帯域:above voice-band)(DSL)の両者の機能性を支援するモデムを提供する。好ましい実施例は、DSPを使用して、音声帯域又は高音声帯域モデムソフトウェアのどちらかを、別個または組合せのアナログフロントエンド、及び共通のホストインタフェース(直列又は並列)と組合せて走行させる。同一の内部構成部分を音声帯域又は高音声帯域のために使用してもよく、またモデムは、電話機により使用するため音声帯域の分離のため内部スプリッタを備えてもよい。

【0026】本発明は、異なる現存するADSL回線コード、離散マルチトーン(DMT)及び搬送波無しAN/PM(CAP)が同じハードウェアプラットフォーム上で音声帯域モデムとして実行を可能にするプログラマブルデジタル信号処理装置(DSP)実施方法を提供する。DSP実施について、回線状態及びサービス費用の要求事項を満たすため望ましい伝送速度も実時間で交渉できる。

【0027】この回線コード及び伝送速度の交渉プロセスは、各通信セッションの初めに、両端にあるモデム間でトーンを交換して実施できる。4ステップ中間帯域デジタル加入者回線(MDSL)モデム初期化プロセスは、回線コード及び伝送速度の両立性のため使用される。

【0028】デジタル加入者回線(DSL)シグナリングは、電話会社中央局(CO)を住宅の加入者に接続する現存する撚線銅電話回線として運ぶために使用されるが、従来のDSLデータモデムは、あるパーセンテージの住宅の加入者に規定のデータ伝送速度で、サービスを提供するように設計される。本発明の新しい交渉方法は、可変伝送速度DSL(VRDSL)システムを可能にする。新しい交渉方法を使用して、可変伝送速度システムは、回線状態、計算能力、ネットワークアクセス可能性、及びアプリケーション要求事項に基づいて、そ

のスループットを適合させる。このサービスは、普通の古い電話サービス(POTS)を損なうことなく電話加入者回線に付加できる。従つて、音声帯域モデム接続は、またDSL接続と関係なく利用できる。

【0029】伝送速度の交渉方法は、多重伝送速度を支持するDSLシステムの組織的制御を提供する。データ伝送速度は、モデム費用、回線状態、又はアプリケーション要求事項に依存して変更できる。このモデムは、VDO、テレビ電話、多重ISDNリンク、及び新しいネットワークアクセス・アプリケーションを含む多くのこととなる応用を支持する可変伝送速度リンクとして機能する。特別のDSL接続、利用できる計算パワー、及びどのような特種なアプリケーションプログラム要求事項をも考慮することにより、伝送速度は、交渉方法により適当なレベルに適合できる。この方式は、対称又は非対称データリンクを提供し、また対称及び非対称リンクの任意の混合を要求する同時の応用も支援する。DSL伝送スループットの対称部分の一部は電話呼出し又はテレビ電話呼出しに使用できる。DSL伝送スループットの非対称部分の一部はインターネットアクセス又はVDOサービスに使用できる。伝送速度の交渉方法は、DSLを使用する多くの異なるネットワークアプリケーションを支援する。

【0030】DSLモデムの典型的な実施は、これまでは、加入者とネットワークの間のコネクションレスサービスだけを支援してきた。しかし、DSLは、ローカル中央局で終端するので、電話にネットワークに都合のよいDSLインタフェースが望ましい。多重仮想(virtual)サービス接続の便宜のため、ISDNのDチャンネルに類似のオペレーション/シグナリング・チャンネルが、交換サービス及び制御メッセージに対してこのましい。CO端DSLモデムにおける処理装置は、シグナリングとデータパケットをCO制御チャンネルサーバへ渡す前に、動作メッセージを集めるのに必要である。

【0031】本発明の、DSLモデムは、コネクションレス並びにコネクションによる(交換)サービスを支援する。

【0032】伝送速度交渉の方法は、可変伝送速度が可能であるDSLシステムにも採用されるのが好ましい。一例は、400Kbpsまでの可変上流伝送スループット及び400Kbpsから2,048Mbpsの下流伝送スループットを提供できる可変伝送速度DSL(VRDSL)システムである。(しかし、この発明は、この例のシステムにより与えられる速度内で変化させることに制約されない。)低いスループットでは、悪い回線状態での動作が支援される。低い伝送速度では、また少ない要求アプリケーションに対して安価なモデムの設計を可能にする。これは、本発明の中間帯域DSL(MDSL)思想と一致しており、音声帯域モデムと同じハードウェアプラットフォームを使用して4000kbpsリ

ンクを提供する。高い下流スループット、VRDSLでは、ADSLと両立させることができる。基本的には、VRDSL伝送速度交渉の方法は、個々の回線状態及び処理力に基づいてスループットを最大にできる価格/性能DSLモデムの範囲でサービスする能力を与える。VRDSLシグナリングにおいて、POTSは依然として同じ電話加入者回線を通じて利用できる。

【0033】中間帯域デジタル加入者回線(MDSL)ソフトウェアシステムに対するホストインタフェース要求事項は、また本発明の一部である。このMDSLネットワークカードを制御するためホストPCプラットフォームの下で走行するソフトウェアは、NDIS3.0WANミニポートドライバとして実施され、それは現存するネットワークドライバ及びアプリケーションと共にウィンドウズNT/ウィンドウズ95の元で動作する。

【0034】中間帯域デジタル加入者回線(MDSL)のための回線接続管理プロセスは、MDSL-C(中央局内のMDSL)とMDSL-R(住宅場所内のMDSL)の間の回線接続を管理するための簡単な、効果的で柔軟性あるインタフェースを提供する。MDSLは、4つの異なる回線モードを使用する。単一のリンクを持つ専用回線(LLSL)、多重リンクを持つ専用回線(LLML)、ソフトダイヤルをもつ交換回線(SLSD)、及びハードダイヤルをもつ交換回線(SLHD)である。LLSLに対するホストインタフェースは、3つの異なる回線状態をもつ、線路電圧降下、線切断、及び線接続である。MDSLモデムの内部状態機械は、回線状態を記録し監視し、状態変化を他のMDSL並びにホスト処理装置に通知できる。本発明の、回線接続管理メッセージの交換のために使用されるプロトコルは、簡単な1対1リンク制御プロトコルである。

【0035】MDSLホストインタフェースは、次の基本的機能をもつ。ホストとMDSLの間のコマンド/制御通信、回線接続管理及び送信/受信データパケットである。MDSLホストインタフェースは、簡単な、ユーザに都合の良い、低価格のインタフェースをホストコントローラに提供する。

【0036】好ましい本実施例においては、MDSLのためのホストドライバインタフェースは、ウィンドウズ95/NT環境の下で走行するNDISWANミニポートドライバとして実施される。このソフトウェアは、MDSLネットワークシステムのメディア・アクセスコントロール(MAC)サブレイヤを制御/管理し、そしてNDISラッパ(wrapper)及びアツパーレイヤ・プロトコルドライバ・スタックと動作し、どの様なインターネット・アクセスアプリケーションも分かり易く走行できる。

【0037】本発明は、またMDSLモデムの時間領域等化器を訓練する簡単なアルゴリズムを提供する。同じ

手順により、F E Tフレーム境界もまた確実に検出される。

【0038】

【発明の実施の形態】図1は、本発明のマルチモードモデム100の第1の好ましい実施例のブロック図を示す。図1において、モデム100は、単一の加入者回線（撚線対）140への音声帯域及びDSL帯域の両方のデータ経路を含み、加入者回線140は、電話会社中央局に接続する。音声帯域アナログフロントエンド（V A A F E）110は、音声帯域（30 H zから3.3 K h z）内の周波数で送信及び受信し、これに対し、デジタル加入者回線アナログフロントエンド（D B L A F E）120は、音声帯域（4 K H zより高い）より高い周波数で送信及び受信をする。スプリッタ130は、加入者回線140へ接続し、また受信信号を音声帯域及び高音声帯域コンポーネントへ分離する。音声帯域及び電話において発生するP O T S（普通の古い電話サービス）は、加入者回線へ直接又はスプリッタ130を介して接続してもよい。

【0039】モデム100は、DSL帯域データ経路の一部として、また音声帯域データ経路の一部として単一のプログラマブルデジタル信号処理装置（D P S）150を利用するが、典型的には2つの別個のデータ入力ポートを使用する。一般的に、DSL帯域は、音声帯域データ経路より遥かに高いビット伝送速度をもち、そこで、別個のD S Pポートをもつことは、緩衝されたマルチプレクサをもつ単一のポートを使用するより便利である。この様なマルチプレクサの使用は代替であるが、明らかに本発明の範囲内にある。例えば、DSL帯域オペレーションモデム100は、100 K H zを中心とし全帯域幅が200 K H zより僅かに少ない上流（住宅から中央局）周波数帯域、及び300 K H zを中心とし全帯域幅が200 K H zより僅かに少ない下流（中央局から住宅）周波数帯域を採用してもよく、この周波数割当てはモデム100の完全なデュープレックス動作を与える。一般的に、多重D S Pを、単一のD S Pの代わりに採用し、遂行される機能を増加し、又は性能を増加してもよい。

【0040】モデム100は、多重回線コードから選択することができ、さらに、モデム100は、音声帯域より高い周波数において高ビット伝送速度DSLモデムとして、又は音声帯域モデムとして、同時に又は連続的に、D S P 150により実行されているプログラムを切り替えることにより、遂行することができる。種々の回線コードプログラムは、D S P オンボードメモリ又は図1に示されない補助メモリに記憶される。また、D S P モデムオペレーションのための代わりの回線コード（例えば、CAP又はDMT回線コード）が使用でき、これもまたD S P 150により実行されているプログラムに依存する。

【0041】図2-3は、モデム100のDSLデータ経路部分を示し、これは、アナログからデジタル172及びデジタルからアナログ170変換器、フィルタ174、176、送信ドライバ178、及び受信機増幅器180を含む。図2は、これに加えて、モデムの内部クロックをホスト（又は中央局）からのクロックと同期させる位相同期ループ182を簡明に示す。図3は、バンドパスフィルタを除去し、その代わりに種々の随意のメモリ型を示し、S R A M 184及び不揮発性186の両者は回線コードプログラムを保持することもできる。モデム100が、音声帯域モデムとして動作するとき、スプリッタ130は、音声帯域アナログフロントエンド120に音声帯域周波数を提供する。

【0042】図4は、DSLモードにおけるモデム100のためのD S P ソフトウエアを示し、これには、

(i) D S P のための随意的カーネル（オペレーティングシステム）190、(ii) ホストインタフェース192、(iii) 随意的管理/保守コントロール194、(iv) フレーミング196、(v) 埋め込みオペレーションズコントロール198、(vi) 埋め込みオペレーションズコントロールをデータストリームでマルチプレクスするためのチャネルマルチプレクサ199、(vii) ビットストリームのスクランプリングのためのスクランブラ論理191、(iii) ビット記号変換、等化、エコー打ち消し、を含むCAP又はDMT論理の様なトランシーバ論理193、及び(ix) 変調器/復調器195論理及び随意的のフォーワード誤り訂正（F E C）が含まれる。

【0043】図5は、ホストとインタフェースするモデム100上を走行するアプリケーションのためのソフトプロトコル階層を示す。物理的層185（層1）は、変調のためのD S P ソフトウエア、ゼットストリームスクランプリング、データストリームをもつマルチプレクシング制御信号を含む。D S P 内のデータリンク層187（層2）は、埋め込まれたオペレーションズコントロール及びフレーミングを含む。ホスト内のネットワーク層189（層3）は、モデムドライバ（例えば、ウインドウ95/NTのためのN D I S 型及びP P P（2点間プロトコル）の様な輸送プロトコル）を含む。インターネットブラウザの様なアプリケーションは輸送プロトコルと相互作用する。

【0044】音声帯域モードの動作のため、モデム100は、標準音声帯域モデム（例えば、V. 34等）い類似のソフトウエアを使用してもよい。

【0045】本発明は、7000 mまでの標準電話撚線対回線上で使用するための新しい高速モデム100を提供する。この新しいモデム100は、M D S L、中間帯域デジタル加入者回線と呼ばれる。M D S L モデム100は、下流及び上流へ送信される信号を分離するため周波数分割マルチプレクシング（F D M）を使用す

る。NDSLのための変調方式は、任意であるが、採用してもよい2つの特定の変調方式は、QAM/CAP及び離散マルチトーンである。中央局(CO)にあるモデムと遠隔ユーザ(RU)端にあるモデムとの間の同期を達成するための立ち上げ手順は、本発明の一部として提供される。

【0046】NDSLモデムの1つの実施例のために選択される変調方式の1つは搬送波無しAM/PM(CAP)である。CAPは、従来の直角振幅変調(QAM)の特殊な場合と考えられる。主な違いは、CAPはその処理を通過帯域で行うが、QAMはその処理をベースバンドで行うことである。

【0047】CAPは、同期のために別個のトーンを使用しない。同期は、送信されたデータ信号を直接使用して達成される。立ち上げ時に、実際のデータが送信される前に、CAP内の等化器を訓練するため特別のデータシーケンスが使用される。

【0048】一つの実施例では、16Kbps-384Kbpsの上流速度(MDSL-CからMDSL-Rへ)及び384Kbps-2.048Mbpsの下流速度(MDSL-CからMDSL-Rへ)を達成するため搬送波無しAM/PM(CAP)変調及び離散多重トーン変調を同じDSPプラットフォーム上で使用する。MDSL-Rのローカル地域ネットワークへのアクセスを可能にするためMDSL-Cもゲートウェイ又はルータとして設置できる。MDSLの応用の例は以下に記載する。

【0049】MDSLハードウェアの原型は、PC又は他のプラットフォームの中へ直接プラグインできるISAカードの上に作られた。この原型は、次のコンポーネントを含む、変調/復調を実施するためのTMS320C541、HOSTとのフレミング及びインタフェースとなるネットワーク物理層、16ビット幅のEEPROM及びRAM、サンプリング速度、解像度、及びMDSLの実施に必要な特性を支持できるD/AとA/Dの組合せ、POTSへの接続に要求されるアナログフロントエンド回路、及びISAバスインタフェース回路。

【0050】図6は、中央局220内の他のモデム100と通信している家庭210内のモデム100を示す。この中央局220モデム100は、種々の能力と負荷を持ち、また加入者回路140は、特別の状態にあつてもよく、そこで、これらのモデムは、回線コード(CAP、DMT、又はその他)、ビット伝送速度、を選択するため初期化を実行し、また等化器を訓練する。次に、これらのモデムはデータ通信を始める。

【0051】図7-8は、DSLモデムをもつ、代わりに、中央局の加入者回線への接続を示し、各加入者回線は、DSL AFE(アナログフロントエンド)をもち、そしてアナログスイッチはAFE出力を、住宅モデ

ム内のDSPに類似のDSP又は多重AFEのための単一のDSPであるDSL処理装置に接続する。中央局は、AFE出力を監視し、そしてデジタルスイッチは、対応する住宅DSLモデムと通信するため利用できるDSPを割当てる。中央局は、住宅内の活動しているモデムを見付けるため、それぞれのAFEを調査する。図7-8に示す様に、中央局DSLモデムは、ローカル地域ネットワーク上の遠隔のアクセスサーバにバケット化情報(例えば、インターネット)を接続し、または、広域ネットワークに一定ビット伝送速度データを接続し、これは直接に、公衆交換電話網中継線を通じて送られる。住宅モデムにより送られる情報は、オンフック信号ではなく、帯域外シグナリング方法(例えば、ISDN Q.931シグナリングに類似)を経由して識別またはシグナリングされ、これに加えて、電話番号が音声帯域でアナログ交換及び回線カードに送られる。図8は、中央局DSLモデム(DSL帯域は既に音声帯域から分離されている)の主要な機能的なブロックである、AFE240、DSP260、通信コントローラ280及びARM又はRISC処理装置290を示す。このモデムは、時分割多重(TDM)バスへ送られている一定ビット伝送速度の伝送データ(音声、テレビ会議等)と制御バス(そして、次に中継線)へ送られているバケット化データ(例えば、インターネット、イントラネット、私設ネットワーク等)の両方への接続をもっている。図8は、用語「xDSL」を示すが、これはADSL又はどのような型のDSLモデムでもよい。これらの種々の機能は全部単一のDSP260に設けることができる。

【0052】AFE240は、中央局220から分離して、中央局へ光ファイバ又は同軸ケーブルを経由して接続されるベデスタルに置いてよく、各ベデスタルは、一束の加入者回線から、1.7km以下の様な短い距離で住宅の所でタップをとる。このやり方で、長い加入者回線に対して高い周波数での減衰を避けることができる。

【0053】中央局に対する、代替のやり方は、音声帯域より高い周波数でDSLモデムが各加入者回線を監視し、そして回線が動作していれば、アナログスイッチがその加入者回線を中央局のDSLモデムに接続する。これは、簡単な監視とアナログスイッチが、AFE監視とデジタルスイッチを置き換えていることを除いて図7を真似ている。同じ方策を、住宅DSLモデムから中央局端(物理的に遠隔ベデスタルに位置する)上のAFEへの加入者回線距離を短くするため用いてもよい。

【0054】図9は、標準プロトコルスタックを持つウインドウ95(又はウインドウNT)を走行させるパーソナルコンピュータ310におけるモデム100をもつシステムを示し、このモデムは加入者回線140を通じて中央局220の対応するモデム100と通信し、

これはイーサネット (Ethernet) (10/100ベースT) インタフェースを経由してインターネットアクセスサーバーに接続してもよい。モデム100は、モデム100のDSL部分がDSL部分を通してインターネットに接続すると同時に、POTS又は音声帯域モデムの両者が他の音声帯域モデムと通信するのを可能にする。

【0055】同様に、図10は、ローカル地域ネットワーク (LAN) 320のためのルータ320として動作するDSLモデムを示し、これは対応するDSLモデムをもつデバイス340、342、344に結合される。

【0056】図11は、PC350内のモデム100に基づくテレビ会議システムの半分を示す。各テレビ会議端は、385+16Kbpsで中央局220内のモデムと通信するモデム100をもつ。中央局モデムは、データを、集信装置とパケタイザ360の間に送信し、パケタイザは、16KbpsシグナリングチャネルをISDNに似たシグナリングメッセージに変換し、そして384Kbpsストリームを、公衆交換電話網を通じてT1/T3サービスに与える。受信側の中央局220は、これらの動作を反転して受信モデム100に供給する。相反する2つの方向へのトラフィックは同様に進行する。POTSは、会議における音声のためにこれらのモデム100について同時に使用できることに注意のこと。ビデオと同期させるためPOTS出力にアナログ遅延を挿入できる。

【0057】図12及び13は、ISDN型シグナリングプロトコル及びメッセージを示し、モデム100は音声又はデータを公衆交換電話網を通じて送る。SS7ネットワークは、ネットワークを通じて呼び設定及びティアダウン (tear-down) のためのISDNユーザ側 (ISUP) メッセージ運搬するための主体となる。

【0058】図17は、マルチモードモデム500を示し、これはモデム100の特徴の、両方のDSL AFE110とVB AFE120、加入者回線140接続のためのスプリッタ130、ISDN回線142へ接続のためのISDNフロントエンド510、これに加えてスピーカ146を駆動し、またハンドフリースピーカフォンを支持するのに使用されるマイクロフォン144の出力を受けるオーディオフロントエンド520をもつ。外部RAM530は、不揮発性 (EEPROM又はフラッシュEEPROM) 及び/又は揮発性 (SRAM又はDRAM) でよい。外部RAM530は、DSP150により使用される異なる回線コードに対する種々のプログラムを含んでも良い。この様な回線コードは、QAM、CAP、RSK、FM、AM、PAM、DWTM等である。

【0059】モデム100の送信部分は、同位相及び直交通過帯域デジタル整形フィルタからなり、これはQ

AMトランシーバ論理の一部として実施され、また受信部分は、QAMトランシーバ論理の一部として実施される同位相及び直交フィードフォワードフィルタ及び交差結合された帰還フィルタをもつ僅かに間隔を置いた複合判定帰還等化器 (DFE) からなる。随意には、QAMトランシーバ論理は、ビタービデコーダを含んでも良い。

【0060】モデム500がアクティブの時は、モデム500は、音声帯域モデム機能性、DSL帯域モデム機能性、ISDN機能性、音声機能性、他の回線コード機能性、等又はこれらの組み合わせを提供してもよい。

【0061】本発明は、多重状及び異なるモデムが同時に単一のDSPハードウェアに実施されるシステムも含む。例えば、音声帯域 (例えば、V.34)、DSL、ケーブル、地上及び他の無線、及び/又は衛星モデムが同時に同じDSPデバイスにより実施される。これは今や、DSPデバイスの処理能力の増加により可能になっている。この方策の利点は、システムが多重モデム (例えば、遠隔アクセスシステム (RAS)) を要求する場合に全体のシステム費用を減少させる。即ち、処理オーバーヘッドの減少に起因して処理要求が減り、またプログラム及びデータメモリバッファの共用によりプログラム及びデータメモリが減少する。例えば、多重状モデムが同時に単一のDSPデバイスにより実行されると、プログラムメモリが減少する。インタフェース及び他の雑多な集中論理は、同じ論理を多重モデム間で共用することにより減少すると共に、統計的多重及び伝送速度制御もさらに便利になる。

【0062】近い時期に、次の状況が優勢になるが、これらの組み合わせは、半導体工業における自然の進歩につれてDSP MIPS能力が増加するに従い拡大するであろう。これらは、同じDSP内の多重音声帯域モデム、同じDSP内の音声帯域及びDSLモデム、同じDSP内の音声帯域及びケーブルモデム、同じDSP内の多重DSLモデム、同じDSP内の多重ケーブルモデム、及びこれらの組み合わせである。

【0063】図18は、音声帯域と高い周波数のDSL帯域の分離のための受動スプリッタ回路を示す。このスプリッタは、またインピーダンス整合を行い、またPOTSに対する受入れ可能なリターンロス値を確保する。

【0064】図19を参照すると、電話機212とモデム500の、加入者回線140を経由しての中央局220への相互接続の略図が示される。

【0065】DSL技術に基づき、今日利用できるシステムは、ISDN基本速度アクセスチャネル及びリピータレスT1である。開発中のDSLシステムは、非対称デジタル加入者回線 (ADSL)、対称デジタル加入者回線 (SDSL)、及び極めて高ビット伝送速度デジタル加入者回線 (VDSL) である。DSLシステムの伝送スループットは、回線損失、雑音環境、及びト

ランシーバ技術に依存する。雑音環境は、自己又は他回路の近端漏話 (NEXT)、遠端漏話 (FEXT)、及び背景白色雑音の組合せである。図20は、多重加入者回線140を示し、近端漏話 (NEXT)、遠端漏話 (FEXT) の発生を略図示す。

【0066】ISDN基本速度アクセスチャンネルのためのDSLの伝送スループットは、160Kbpsである。搬送波無しT1のためのHDSLの伝送スループットは、800Kbpsである。ADSLの伝送スループットは、上流 (加入者から電話中央局) において16Kbpsと640Kbpsの間であり、下流においては1.544Mbpsから607Mbpsの間である。MDSLのスループットは、現在、上流において384Kbps、そして下流においては384Kbpsから2.048Mbpsの間であると信じられる。

【0067】通過帯域DSLシステムは、直角振幅変調 (QAM) 又は搬送波無しAM/PM (CAP) 回線コードを使用する単一の搬送波で実施される。単一の搬送波システムは、チャンネル歪みを補償するための適合チャンネル等化器に依存する。チャンネル等化器は、通常シグナリングボー速度の倍数で動作する。図21はCAPトランシーバのブロック図を示す。

【0068】より詳しくは、D/A614は、送信機フィルタ610、612及びフィルタ616に接続される。フィルタ616は、チャンネル620に接続される。チャンネル620は、フィルタ630に接続され、これはA/D632に接続される。A/D632は、等化器634、636に接続される。回路638の一部は、時間を回復する。

【0069】DSLシステムは、離散マルチトーン (DMT) 回線コードを使用する多重搬送波で実施される。DMTシステムは、チャンネルを多くのサブチャンネルに分割し、チャンネルの容量をより良く使用したチャンネル歪みを減少しているが、これは、比較的簡単な適合チャンネル等化器の使用を可能にし、この等化器はチャンネルのインパルス応答の時間の広がりを修正するよりは圧縮するだけである。DMTサブチャンネルのシグナリング帯域速度は単一の搬送波システムの帯域速度より遥かに低い。

【0070】図22は、DMTトランシーバのブロック図である。より詳しくは、IFFTブロック640は、D/A644に接続され、後者は送信フィルタ646に接続され、これはチャンネル650に接続される。チャンネル650は、フィルタ660に接続され、後者はA/D662に接続され、これは等化器664に接続され、後者はFFTブロック666に接続される。立ち上げ642及び時間回復668回路も含まれる。

【0071】一つのMDSLモデムの実施例では、ハードウェア費用及び漏話雑音レベルを低下させるために、周波数分割完全デュプレックスを使用する。この様な、MDSLモデムは、中央局と加入者の間に7kmまで

の回線長さに対して最少384Kbpsの完全デュプレックス伝送リンクを提供するであろう。良好な加入者回線状態の下では、このMDSLモデムは、より高い伝送スループットを与えることができ、これはチャンネル容量又は加入者端モデムのハードウェア能力により制限される。加入者端MDSLモデムの完全な特徴バージョンは中央局端のADSLモデムと通信する。MDSLモデムの送信機及び受信機部分は、CAP又はDMTのどちらかの回線コードを実施できる。

【0072】図23は、MDSLモデム600のブロック図を示す。モデム600は、D/A674に接続される送信機676をもち、D/A674はフィルタ672に接続され、後者はハイブリッド回路670に接続され、後者はスプリッタ130に接続される。ハイブリッド回路は、またフィルタ678に接続され、後者はA/D680に接続される。A/D680は受信機682に接続され、この出力は受信信号である。タイミング回復ブロック684が、中央局クロックタイミングを回復するため使用される。

【0073】初期化プロセスの目的は、中央局220及び加入者端210での電話加入者回線のMDSL能力を確認することである。初期化プロセスは、チャンネル620を調べ、そしてトランシーバ訓練に役立つ情報を作る。このプロセスは、次に、多重選択が利用できると仮定すると、回線コードを選択し、そして、チャンネル制限、トラフィック状態、又は使用料金に基づき、伝送スループットを交渉する。

【0074】後で述べる初期化プロセス、チャンネルの調査、回線コードの選択、伝送速度の交渉、及びトランシーバの訓練である。

【0075】加入者端のMDSLモデムはある期間、上流へ調査トーンを送るが、これは、これらのトーンの一部に位相変化があつてもなくても、所定の時間シーケンスに従う。第1の期間の後中央局端のMDSLモデムは、上流中のチャンネル調査トーンに応答するがこのときも、これらのトーンの一部に位相変化があつてもなくてもよい。この最初のチャンネル調査期間は、もし希望し、又は必要であれば繰り返しても良い。

【0076】最初のチャンネル調査期間の後、加入者端のMDSLモデムは、中央局端のモデムの回線コードの能力を決定し終わっており、また下流帯域に対するチャンネルモデルをもっており、同様に、中央局端のMDSLモデムも、加入者端のモデムの回線コードの能力を決定し終わっており、また上流帯域に対するチャンネルモデルをもっている。

【0077】チャンネル調査期間の後、加入者端のMDSLモデムは、その回線コードの能力/優先を、所定の期間シグネチャトーンを送ることにより指示/確認すべきである。同様に、中央局端のMDSLモデムも、回線コードの選択を、所定の期間シグネチャトーンを送ること

により応答/確認すべきである。このシグネチャトーンの交換は、特定の回線コードの選択を決定するため、ある制限された回数繰り返すことが好ましい。

【0078】別の組のシグネチャトーンが、両端のMDSLモデム間で、伝送速度の交渉のため交換される。加入者端のMDSLモデムは、その速度能力及びその優先を送る。中央局端のMDSLモデムは、その能力及びその速度選択で応答する。これらのMDSLモデムは、後述の、所定の速度変更手順で速度選択を決定する。加入者端の伝送速度の優先は、回線状態、ハードウェア能力、及びユーザの選択又はアプリケーション要求事項に依存する。中央局端の伝送速度の優先は、回線状態、及びトラフィック負荷に依存する。回線状態の変化及びユーザの選択により通信セッションの間の伝送速度の変更が許されるのが好ましい。

【0079】速度の交渉の後、両端のMDSLモデムは、従来の方法に従いトランシーバ訓練を開始する。異なる回線コードに対しては、異なる時間領域訓練シーケ

$$v_i = \frac{b_i}{a_i} e^{-j(\phi_i - \psi_i)}$$

【0083】等価チャネルのインパルス応答は、次式の急速フーリエ変換で計算できる。

$$h_k = \sum v_i e^{-j2\pi\Delta f T i k}, \quad k = 0, \dots, m$$

【0085】ここで、Tは、サンプル期間である。周波数 Δf は、チャネルインパルス応答に依存する。 n 個のサンプリング期間のチャネルインパルス応答は、次式で表わされる。

【0086】

【数3】

$$\Delta f \leq B / n = N\Delta f / n$$

【0087】ここで、Bは、全帯域幅である。2つの異なる回線コードを区別するため、隣接する2つのトーンの位相は、回線コードの一方に対して180度だけ逆転してもよい。この回線コードはDMTである。チャネルの歪みの後、異なる回線コードを識別するため、次式のものを選ぶ。

【0088】

【数4】

$$\Delta f = \frac{B}{2n}$$

【0089】30個のサンプル及び100KHzの帯域幅に対して、次式のものを選び、そして、Nは64とする。

【0090】

【数5】 $\Delta f = 1.7 \text{ KHz}$

【0091】チャネル調査トーンは、少なくともチャネルの幅がりの2-3倍とする。位相の変化の可能性から、チャネル調査トーンの持続時間は、チャネルモデル

ンスが使用される。トランシーバ訓練プロセスの速度を上げるため、チャネル調査中に得られたチャネルモデルの使用は随意である。

【0080】上流および下流の調査トーンのスペクトルが図24に示される。上流CAPトーン690及び下流CAPトーン692は左側に示され、他方上流DMTトーン694及び下流DMTトーン696は右側に示される。DMTスペクトルの「破線」は位相シフトを示す。

【0081】簡単のために、全部の周波数トーンは、周波数 $i\Delta f$ 、振幅 a_i 、及び位相 ϕ_i （通常それは0又は π ）で等しい間隔であると仮定する。受信機においては、受信トーンの振幅及び位相を検出してよい。 i 番目の周波数トーンを検出された振幅及び位相は、 b_i 及び ϕ_i である。 N 個の調査トーンがあると仮定すると、周波数 $i\Delta f$ でのフィルタを含む等価チャネルの周波数応答は、次式で表わされる

【0082】

【数1】

$$i = 0, \dots, N - 1.$$

【0084】

【数2】

の回復に必要な時間の4-10倍とすべきである。

【0092】 N 個のトーンを使用すると、一定のトーンをもつ単位時間に $M = 2^N$ の異なるメッセージを表わすことができる。使用されるトーンの数に対して利用できる単語は指数的に増えるので、有効なメッセージは小さな組のトーンで送ることができ、例えば、2、3、又は4個の異なる周波数だけでよい。次ものは、メッセージのリストの例である。

384Kbps/CAP

768Kbps/CAP

1.544Mbps/CAP

2.048Mbps/CAP

384Kbps/DMT

768Kbps/DMT

1.544Mbps/DMT

2.048Mbps/DMT

最高速度を好む

最良価格を好む

パケット多重許可

低速のみ利用可能

【0093】トーンは、DMT回線コードに対して使用されるようにIFFTの動作により発生できる、単位大きさとゼロ/180度位相ベクトル信号が、チャネル調査目的でIFFT動作に供給される。選択されたゼロ位相ベクトルは、シグネチャトーンの発生のため使用され

る。

【0094】トーンは、DMT回線コードにも使用されるようにFFT動作により再生される。各トーンの振幅及び位相情報は、複素ベクトルとして再生される。ランダムサンプリング位相ベクトルは、シグネチャートーンの発生のため使用される。トーンは、DMT回線コードにも使用されるようにFFT動作により再生される。各トーンの振幅及び位相情報は、複素ベクトルとして再生される。ランダムサンプリング位相に起因する共通位相差は計算できる。補償により複素ベクトルが作られ、これは、次にチャンネル伝送スループット及びチャンネルインパルス応答の計算に使用され、これはトランシーバ訓練に使用される。

【0095】もし、MDSLサービスが、電話回線を通じて利用できれば、中央局端のMDSLモデムは、トーンの調査の為に「オン」で、上流周波数帯域を監視すべきである。

【0096】一旦電力が「オン」でユーザのサービス要求がなされると、加入者端のMDSLモデムは、調査トーンを所定期間、上流へ送り、そして次に下流の調査トーンを監視する。中央局端のMDSLモデムは、調査トーンを検出し、ランダム位相を補償し、これらを記憶し、そして上流チャンネル伝送スループットを計算する。そのうち中央局端のMDSLモデムは、下流周波数帯域へ調査トーンを送る。

【0097】加入者端のMDSLモデムは、調査トーンを検出し、ランダム位相を補償し、これらを記憶し、そして下流チャンネル伝送スループットを計算する。加入者端のMDSLモデムは、回線コード及び伝送速度優先を示すため、シグネチャートーンを上流帯域の中に送る。

【0098】中央局端のMDSLモデムは、シグネチャートーンを検出し、その好ましい提供物に対応するシグネチャートーンで応答する。加入者端のMDSLモデムは、そこで提供物を確認し、又は提供物の修正を要求するためシグネチャートーンを送る。これらのMDSLモデムは、モデムの提供物の確認の後、トランシーバの訓練期間に入る。

【0099】DSL通信チャンネルのスループット容量は、回線状態及び／又はネットワークのアクセス可能性で変化するであろう。回線状態は、COと住宅の間の物理的接続の達成可能なスループットを示す。ネットワークのアクセス可能性は、DSLチャンネルを主体ネットワークへリンクするサービス提供者の接続の能力を示す。発明の速度交渉方法は、DSLシステム的能力制限要因の詳細な理解を組み込んでいる。

【0100】DSLシステムは、伝統的に、サービスを提供すべき最悪の場合の回線状態に対して設計されている。この方策は、電話会社に対する一般的な設備手順を簡単にしている。しかし、DSL伝送スループットを最悪の回線状態において達成されるスループットに制約す

ることは、大部分のDSLシステムを、これらの潜在能力より下で十分動作させている。発明の方法は、各ローカル回線の物理的伝送スループットを最大にするための組織的手順を提供し、大部分のDSLシステムが、伝統的に設計されているよりも、遥かに高い速度での動作を可能にしている。事実、この方法は、大多数のDSLモデムが、モデムハードウェアの能力により制限されるだけの伝送スループットを達成可能にしている。速度交渉方法はまた、回線状態又はネットワークのアクセス可能性で変化するときに、最高の可能なスループットを維持するため時間変化する適合性を提供する。

【0101】然系対DSLチャンネルは、雑音と干渉の存在の下で送信信号を確実に区別する受信機的能力により制限される。最大の可能なスループットは、その上限は、図16に示すような、物理的リンクの理論的チャンネル容量である。リンクのチャンネル容量は、使用される帯域幅、受信信号特性、及び雑音と干渉により決定される。速度交渉方法は、極端に長い電話加入者回線により支持される低速の選択を提供し、他方、高いスループットを達成するため、短い回線でDSLモデムの動作を可能にする高速の選択を提供することによりDSLの能力範囲を増加する。

【0102】速度交渉方法は、DSL伝送媒体の動的性質を考慮する。DSLは、時間変化するチャンネルで、その容量は、改善する／悪化するチャンネル状態に起因して変化する。チャンネル状態が変化すると、理論的最大スループットもまた変化する。チャンネル特性の時間変化する性質は、チャンネルをある時間にわたり最大の効率で使用するための速度交渉技法の必要を示している。この技法は、困難なチャンネル特性の間スループットを低下させることにより、DSL接続を維持する能力を与える。これはまた、良好なチャンネル特性の期間の間、モデムがスループットを増加させ、また接続を最良に使用することを可能にする。理想的には、各端でのトランシーバが、チャンネルの監視ができ、状態が変化するとき、それらのスループットを最大にすることである。実際の送信機／受信機は、利用できる容量、利用できる信号処理資源、及び特別のアプリケーション要件に基づき物理的チャンネルのスループットを増加し、又は減少するように設計できる。数個の速度適合方法が存在（例えば、標準CCITT V. 34音声帯域モデム標準）するが、2つの特に便利な技術につき、2つの明確に異なる変調方法に対して後で述べる。しかし、速度適合の技術は、容易に他の変調及びコード化方式に拡張でき、そしてこの様な拡張は本発明の一部と認められる。

【0103】ネットワークのアクセス可能性とは、この記載では、データのローカル回線から主体ネットワークへの転送に関連する速度及び／又は遅延を示す。この測度は、使用される特定の、主体ネットワーク（例えば、インターネット、ATM等）、サービス提供者により与

えられる帯域幅、及びネットワークトラフィックにより影響されるかも知れない。この発明により定義される技術は、特定の主体 (backbone) ネットワーク上の使用に制約されない。

【0104】VRDSL接続はある伝送スループットの能力はあるが、全スループットは、ときには対応するCO主体ネットワークには接続されない。PSTN (公衆交換電話網) を通るVRDSLを備えるサービスに対して、接続は、サービスが開始されるときだけに行われたい。インターネットアクセスの様なローカルCOであるVRDSLを備えるサービスに対しては、あるスループットをもつ専用回線、又はダイヤルアップ回線接続が、好ましい費用構造に依存してなされる。各VRDSLモデムへの利用できるCO主体ネットワークは、異なる時間では異なることがある。加入者所望のスループットもまた、アプリケーションが異なると変わることがある。

【0105】VRDSL物理的伝送リンクにより与えられるより低い実際のスループットをもつトラフィック集中がCO主体ネットワークにおいて実現できる。統計的多重も、各VRDSLモデムに対して別個のアナログフロントエンドを使用することにより実現できる。対応するデジタル部分の要求される数は、トラフィックの反応に依存して少なくできる。極端な場合、COのVRDSLモデムデジタル部分は、トラフィック指示チャンネルとして音声帯域を用い、またモデムのデジタル状態部分のコピーをRAM内に保持することにより、アクティブVRDSLリンクの中へ多重化できる。

【0106】VRDSL通信モデルが図25に示される。このモデルの唯一の目的は、開示された速度交渉技術の理解を助けることである。このモデルは、別個の住宅7210と中央局7220からなり、これらは、機能的にわかれた層として表示される。住宅端末7210の機能性は左に示される。最下層7330は、通信ハードウェア層で、これは、変調器/復調器、信号コンディショニング、タイミング、同期化、および誤り訂正コーディングである。この層は、またデータポンプ層とも呼ばれる。第2の層7320は、ハードウェア制御層である。この層は、下層により受けたデータをよくまとめる、フレーミング制御及び他のデータパッケージ機能をもつ。第3の層7310は、ソフトウェアドライバ層である。この層は、ハードウェアレベルと住宅で走行するアプリケーションプログラムとのインタフェースとなる。第4の層 (トップ) 7300は、アプリケーションソフトウェア層で、これは、住宅で走行するアプリケーションプログラムにより与えられる全部の機能を含む。この層は、異なる同寺アプリケーションに割当てられた管理するソフトウェア並びにアプリケーション自体を包含する。従来のソフトウェア・アプリケーションプログラムは、チャンネルを要求し、下層により与えられるスループットを受け入れる (交渉なし)。将来のソフトウェ

ア・アプリケーションプログラムの発生は速度交渉の要求及び能力を持つかもしれない。

【0107】このモデルのCO7220部分も4つの層をもつ。下部の3つの層、7430、7420、7410は、モデルの住宅側ときわめて類似している。(しかし、実際の実施は著しく異なる。) COにおける第4 (トップ) 層7400は、ネットワークアクセスソフトウェア層と呼ばれる。この層は、DSL接続と主体ネットワークのインタフェースに要求される機能を備える。

【0108】速度交渉方法において、モデルの各層は、下層及び上層と通信し相互作用する。層間の通信のためのプロトコルは定義される。図25に示す様に、速度交渉を開始するため、層は下層にR (速度要求) を指示する。「R」は、図25では対応する下向きの矢印で示す。下層は、A (利用可能速度通知) を上層に指示し、利用可能速度を通知する。図25では対応する上向きの矢印で示す。R及びAの意味は、異なる層インタフェースでは異なるが、交渉のプロセスは同様である。

【0109】速度表 (rate table) は、層間のR及びAシグナリングに対する共通のシンタックス (syntax) として定義される。速度表は、特定の層が取得しようと試みる事の出来る速度を定義する。

(一般に、これはモデムのハードウェア限定事項として定義されるであろう) 速度要求 (R) の間、上層は下層に、速度構造の変更の希望を合図するかもしれない。もし、下層が、それ自身を新しい一組の動作パラメタに再構成し、要求された速度の達成が可能であれば、その様にして、これを上層に指示する。もし、下層が、要求された速度の受入が出来ないと決定すれば、上層は、現在の動作状態において利用可能な速度についての情報と共に通知される (A)

【0110】下層は、また、もしより低い又はより高い達成可能なスループットに起因して、動作状態が変化すれば速度交渉を開始できる。上層は、そこで、新しい動作状態に基づいて、速度要求で応答することができる (R)。

【0111】この共通層インタフェースは、速度交渉方法を簡単にする。各層インタフェースで速度パラメタは異なるが、相互作用方法は同様である。

【0112】各層は、概念的に、その層とDSLの他方端の対応する層と間の通信リンクと見ることができる。住宅及び中央局における対応する層を接続する回線により示される様に、

1. 住宅及びCOにおける通信ハードウェア層7330、7430は、仮想でない「生」の接続である。これは、物理的接続であり、これを通じて実際の変調が起こる。
2. ハードウェア制御層7320、7420は、通信リンクを仮想の「訂正された」データストリームと見ることができる。これは、物理的タイミング、同期、制御、

及び誤り訂正コード化冗長記号が除去された、後のチャネルの実際のスループットである。

3. ソフトウェアドライバ層7310、7410は、接続を、データリンクチャネル(DLC)と呼ばれる仮想のチャネルと見る。便利のために、DLCは、多重N kbit/secチャネル(N=16又は64)を表わすフレーム構造もよい。この制御チャネルは、下層チャネルに埋めこまれるか又はDLC接続から完全に分離してもよい。例えば、制御シグナリングは、v. 34モデム接続を経由する音声帯域で実施してもよい。

4. 応用ソフト層7300は、CO内のデータ提供位置又は主体ネットワークへの仮想「アプリケーションリンク」7500を見る。

【0113】速度適合の基本的要件は、速度表、DSL通信モデルの上層へ通信できる良く定義された一組の達成可能な速度である。速度表は、接続の両端のハードウェアの能力により決定される。立ち上げ又はリセットの間に、1対のモデムは、両者が支持できる速度表の記入事項(エントリ)について同意しなければならない。与えられたチャネル状態の下での許された速度は、次に表内の正当な状態として代表される。モデルのそれぞれ異なるレベルは、他の層における詳細を気にすることなく速度表シンタックを経由して通信できる。この速度表は、1つの変調及び/又はコーディング方式毎に本質的に変わっているが、チャネル状態に依存して速度が許可及び不許可になる構想は変わらない。

【0114】次に、ハードウェア制御層7320、7420と通信ハードウェア層7330、7430との間の速度交渉が、本発明に従ってどの様に行われるかを説明する。変調パラメタは、種々の速度を受け入れるため変化でき、それぞれの層はこの速度を使用して相互作用する。次に、2つの可能な、変調を基にした速度適合技術及びDSL通信モデル内の下部の2層間で共有される速度表の例を説明する。

【0115】デジタルデータの高速直列伝送の場合において、デジタル記号が、ある数のビット、例えば、Nを表わすため選ばれる。Nビットからなるそれぞれの群は、記号でマツプされ、これはチャネルを通じて伝送される。デコーダにおいて、伝送された記号を決定するため判定がなされる。もし、正しいとの判定があると、伝送されたビットは正しくデコードされる。

	R = 1e5	R = 2e5	R = 3e5	R = 4e5	R = 5e5
N = 2	200 kbits/s	400	600	800	1000
N = 3	300	600	900	1200	1500
N = 4	400	800	1200	1600	2000
N = 5	500	1000	1500	2000	2500

【0122】表1における、速度Rは、記号/秒の単位であり、表では簡明のため科学的表示法で示す。表中の記入事項(エントリ)は、各記号で表わされる与えられた速度R及びNビットに対する達成可能な伝送スループ

【0116】スループットを変更する方法では、各記号で表わされるビットの数をえ、一方記号速度は一定に保つ。各記号で表わされるビットの数を増加すると、伝送されるビットの数は増え、しかし雑音の免疫は低い。各記号で表わされるビットの数を減少させると、雑音の免疫は増し、伝送の強さは改良されるが、低いスループットが代価である。帯域幅は、いずれの場合も同じままである。

【0117】スループットを変える別の簡単な方法は、伝送チャネルにおいて使用される帯域幅を変えることである。帯域幅を拡張することにより、より多数の記号がチャネルを通じて、与えられた間隔で伝送される。記号速度は、ほぼ帯域幅に比例する。しかし、DSLモデムの処理要求事項は、帯域幅と共に増加し、より高い帯域幅は、変調/復調のためのより多くの計算を要求する。最大の使用可能帯域幅は、チャネル状態又はモデムハードウェア処理能力の制約により制限される。

【0118】最初に、通信リンクを記述する一組のパラメタ及びこれらパラメタが取る一組の値を定義する。公称直列伝送速度をRとする。DSLモデルが、変更できる最少の速度ステップをdRとする。もし、最少速度が $R - 2 * dR$ そして最大速度が $R + 2 * dR$ であれば、達成可能な速度の組は、 $\{R - 2 * dR, R - dR, R, R + dR, R + 2 * dR\}$ で与えられる。例えば、 $R = 300 \text{ kilo-symbols/second}$ 、また $dR = 100 \text{ kilo-symbols/seconds}$ である。達成可能な速度の組は、 $\{100, 200, 300, 400, 500\} \text{ kilo-symbols/seconds}$ である。

【0119】各伝送されるデジタル記号により運ばれるビットの数をNとする。例えば、DSLモデムは、組 $\{2, 3, 4, 5\}$ におけるNを持つ動作を支持する。Nのより高い値は、より多くのビットを運ぶが、雑音の許容範囲は低くなる。

【0120】R及びNCの速度パラメタを使用し、またこれらは、独立して上の値を取ると仮定すると、速度表は、次の様に定義できる。この表は、直列伝送(例えば、CAP)速度の例を示す。

【0121】

【表1】

ットを示す。

【0123】離散マルチトーン(DMT)変調は、低速データ記号を並列サブチャネルを通じて伝送する。高速直列データストリームを多重低速データストリームに分

割することにより、多重低速データストリームはそれぞれ別個のサブチャネルで伝送され、システムは周波数選択的チャネルにより良く整合する。全体の帯域幅の良好な部分（高い信号対雑音比（SNR）をもつ）が、より多くの数のビット／記号をもつ記号の伝送に使用される。等しくない数のビットは、各サブチャネルの利用できる容量に依存して、それぞれ異なるサブチャネルに割当てられる。本質的にはデータは、全体の帯域幅を極めて効率的に使用できるやり方で、それぞれのサブチャネルに分配される。

【0124】高速直列データストリームについて、DNTシステムの全体の帯域幅は、全体の所望のスループット、チャネル状態、及びモデムハードウェア能力に従って増加し、又は減少させることができる。これに加えて、DMT変調は、一度に帯域幅を単一のサブチャネルだけ低下又は付加する能力をあたえる。多数のサブチャ

	T = 32	T = 64	T = 96	T = 128	T = 160
N = 2	200 kbits/s	400	600	800	1000
N = 3	300	600	900	1200	1500
N = 4	400	800	1200	1500	2000
N = 5	500	1000	1500	2000	2500

【0127】パラメタTは、サブチャネルの数を表わし、ここで、各サブチャネルは、約3.3KHzの帯域幅をもつ。Nは、全部のサブチャネルにおいて、表わされるビット／記号の平均数を表わす。表の記入事項はキロビット／秒で与えられる。

【0128】実際のDMT速度表では、サブチャネルは、1つの増分で付加又は除去することがある。また各サブチャネルに割当てられるビットの数は独立に制御できる。従つて、DMT速度表は、極めて小さな速度増分での調節の可能性をもつ。

【0129】ソフトウェアドライバ層7310、741

	cr1	cr2	cr3	cr4	cr5
N = 2	192 kbits/s	384	576	768	960
N = 3	288	576	864	1152	1440
N = 4	384	768	1152	1536	1920
N = 5	480	960	1440	1920	2400

【0131】列パラメタは、異なるチャネル資源モード（cr1、cr2... cr5）としてラベルされるが、行パラメタは、各記号により表わされる平均ビット数に対応する。記入事項は、VRDSLモデムにおける「訂正された」データストリームに対する達成可能速度を表わす。

【0132】応用ソフトウェア層7300、7400とソフトウェアドライバ層7310、7410の間の速度調節情報は、速度表又は総利用可能スループットで特定される。簡単のため、ソフトウェアドライバ層は、応用ソフトウェア層に全利用可能速度を示すことができる。応用ソフトウェア層における管理機能は、総利用可能スループットのある部分を種々のソフトウェア・アプリケ

ネルをもつDNTシステムに対して、これは可能な帯域幅の極めて大きな選択を作り出す。もし望むならば、全体の帯域幅は一定にしてサブチャネルの数を変更できる。

【0125】簡単のために、サブチャネル帯域幅が一定のみであるが、使用される全体のチャネルの帯域幅は使用されるサブチャネルの数により制御されるDMTシステムを考える。Tは、伝送に使用されるサブチャネル又はトーンの数を表わすとする。Nは、サブチャネルにまたがるビット／記号の平均数を表わすとする。Nは最早、高速直列伝送システムの時の様に、整数である限定はない。この例のためにしかし、Nはほぼ整数の値であると考えられる。次の表2は、DNTに対する速度表の例である。

【0126】

【表2】

	T = 32	T = 64	T = 96	T = 128	T = 160
N = 2	200 kbits/s	400	600	800	1000
N = 3	300	600	900	1200	1500
N = 4	400	800	1200	1500	2000
N = 5	500	1000	1500	2000	2500

0は、前述のものと極めて類似の速度表の手段によりハードウェア制御層7320、7420と通信する。しかし、表パラメタ及び表記入事項は異なる。同期、復調、誤り訂正デコーディング、およびハードウェア制御ビットの除去の後、結果として得られる速度表は、前述の変調方式のいずれについても次の表3となる。この速度表は、ソフトウェアドライバ層とハードウェア制御層の間の相互作用に対して使用される。

【0130】

【表3】

ーションプログラムに割当てて。次のものは、総データスループットの分割及び管理の構想的見方である。

【0133】データリンク層における速度交渉は、VRDSLにおける次の事象により開始される。

— 新チャネルの要求又は現在のチャネル速度の変更の様な、VRDSLにおける現在のデータ接続又はチャネルの割当ての変更の要求

— 総チャネル容量増加又は減少の総チャネル容量変化をVRDSL物理的層が検出したとき

【0134】VRDSLの初期化の後、制御チャネル（例えば、16kbp/s）は初期のチャネル接続として割当済みである。制御チャネルは、全部の物理的回線接続時間中保持される。速度交渉情報を含む全部の制御情

報の送信／受信が使用される。

【0135】データリンク層における速度交渉は、幾つかの信号データ型式及び有限状態オートメーションとして記述できる。速度交渉信号データは、PPPデータリンク層フレーム構造の情報フィールドの様なデータリンク制御プロトコル内にカプセル封入される。プロトコルフィールドは、VRDSL速度交渉プロトコルに対して型0xc024を指示する。パケット形式は図26に示される。

コード：コードフィールドは、1オクテットで、速度交渉パケットの種類を示すコードI-IIがPPP LCPに対して保留される。VRDSLに対して、次の特別の定義がある。

- 13 チャンネルマツプ変更要求 (request)
- 14 チャンネルマツプ変更Nak
- 15 チャンネルマツプ変更拒否 (reject)
- 16 チャンネルマツプ変更Ack

ID：IDフィールドは、1オクテットで、突合せ要求及び返答を助ける

長さ：長さフィールドは、2オクテットで、全速度交渉信号データパケットの長さを示す。

チャンネルマツプデータ：チャンネルマツプデータフィールドは、2以上のオクテットで、VRDSL回線における現在のチャンネル割当て及びチャンネル変更要求を反映する。それは、それ自身のヘッダ及びチャンネルエントリフィールドで表わされる情報の2部分を含む。

- ー 現在のチャンネルマツプ
- ー チャンネルマツプ変更要求

これらの、情報の2部分は、2オクテットチャンネルエントリフィールドにより記述される。これらを区別する方法は、チャンネルマツプ変更要求に対して、チャンネルエントリの最上位ビットが高に設定される。コードが14

(チャンネルマツプ変更Nak)の時は、チャンネルマツプデータフィールドは、次のものを含む。総容量、利用可能容量、現在のチャンネルマツプ、及びナック(Nak)

された1又はそれより多い記入事項。これらのナック(Nak)されたチャンネル記入事項は、それらの最上位ビット(msb)によりフラグを付ける。コードが15又は16の時は、チャンネルマツプデータフィールドは、次のものを含む。総容量、利用可能容量、現在のチャンネルマツプデータ。

検査合計(Checksum)：検査合計フィールドは、標準TCP/IPアルゴリズムを用いて計算される。メッセージ(検査合計フィールドを除く)における

全16ビット整数の合計の1の補数。リンク層速度交渉は、またVRDSLにおいて、チャンネルマツプ変更(CMC)と呼ばれる。CMC手順は、特定の事象又は行動によりトリガされる状態変化により記述される。図28, 29は、能動(アクティブ)及び受動CMCプロセスの間のリンク層速度交渉のための状態図を示す。

【0136】VRDSL通信モデルに基づき、伝送速度を変更する能力のあるモデムハードウェア、及び可変速度管理ソフトウェア、図30に示す速度交渉方法が採用されてもよい。図30は、全体の速度交渉の簡略な機能図を示す。

【0137】現在のQAMに基づく音声帯域モデムは、呼びと応答モデム間の通信を起動するためハンドシェイク(handshake)シーケンスを使用する。同期を得るため応答モデムは対応する配列(constellation)点の代替の記号を送信する。例として、V.32モデムは、同期プロセスにおいて、図21の配列点A、B、C、及びDを使用する。応答モデムは、256記号の間交番する記号ABABAB...を送信する。256記号の後、交番する記号CDCDCD...が16記号の間送信される。2つの記号シーケンスの間の過度期間は、良く定義された事象を与え、これは呼びモデムにおいて時間基準を発生するのに使用される。第2の記号シーケンスの後、モデムは、両方に知られている記号シーケンスの送信を開始する。このシーケンスは、呼びモデム受信機での等化器を訓練するのに使用される。図31は、V.32訓練配列を示す。

【0138】音声帯域チャンネル(30Hz-3.3KHz)の周波数応答は、名目上平坦である。交番するABAB...及びCDCD...記号は、チャンネルの等化前に確実に検出できる。しかし、これは、NDSLモデムには、当て嵌まらない。1/4T1に対して、モデムは、電話回線の500KHzまでのスペクトルを使用する。図32は、電話CSA回線6の周波数応答を示す。回線の部分的等化を可能にする立ち上げ手順は、タイミング同期が試みられる前に要求される。

【0139】好ましい実施例では、立ち上げ手順を、MDSLモデムに対して使用する。図33は、CAP回線コードを使用するCO及びRUモデムのための、提案されている立ち上げ手順のための時間線を示す。次の表4は、図33の種々のセグメントを識別している。

【0140】

【表4】

セグメント	記号
A, D	<p>1つの直交チャネルは、CAP配列の最大値を使用した反復K記号シーケンスである。16配列に対して、チャネルは± 3の値を取るしかできない。</p> <p>他の直交チャネルは、CAP配列の全可能変を使用するランダムシーケンスである。16配列に対して、チャネルは± 1又は± 3の値を取るしかできない。</p>
B, E	<p>1つの直交チャネルは、長さKシーケンスで、これはセグメントAで使用するK記号シーケンスの反転バージョンである。</p> <p>他の直交チャネルは、CAP配列の全可能変を使用する長さKランダムシーケンスである。16配列に対して、チャネルは± 1又は± 3の値を取る。</p>
C, F	<p>1つの直交チャネルは、長さLランダムシーケンスで、これはCAP配列の全可能変を使用する。16配列に対して、チャネルは± 1又は± 3の値を取る。</p> <p>他の直交チャネルは、長さLランダムシーケンスで、CAP配列の全可能変を使用する。16配列に対して、チャネルは± 1又は± 3の値を取る。</p>

【0141】立ち上げ手順は、次の通りである。

COモデム

1. COモデムは、常に「ON」とであると仮定するが、これは理想的状態である。それは、連続的にセグメントAを送信し、そしてセグメントBを待ち受ける

RUモデム

1. RUモデムは、回線にきて、COモデムからのセグメントAの待ち受けを開始する。

2. 一旦セグメントAを検出すると、それはセグメントDの送信を始める。

COモデム

2. 一旦COモデムが、RUモデムからのセグメントDを検出すると、それはセグメントB, Cを送信し、そしてRUモデムからのそれ以上のハンドシェイキングなしにデータを妥当とする。

RUモデム

3. RUモデムは、セグメントBを待ち受け、一旦検出すると、それはセグメントE, Fを送信し、そしてCOモデムからのそれ以上のハンドシェイキングなしにデータを妥当とする。

4. セグメントBの検出は、同期手順においては臨界的タイミングの瞬間である。それが検出された後、RUモデムは、セグメントCからのデータを使用してその等化

器の訓練を始める。

COモデム

3. COモデムは、RUモデムからのセグメントEを待ち受ける。セグメントEの検出は、同期手順においては臨界的タイミングの瞬間である。それが検出された後、COモデムは、セグメントFからのデータを使用してその等化器の訓練を始める。

【0142】受信機は、初期のタイミング同期を得るため、周期的等化技術を使用する。立ち上げ時に、RUモデムは、K記号期間と時間期間が等しい、僅かに間隔を置いた適合等化器をセットアップするが、Kは、例えば、15でよい。これは同期等化器と呼ばれる。もし、同期等化器が、記号期間の2倍で動作すれば、必要なタップも数は $2 \times X$ である。記号期間当たり4個のサンプルに対して、必要なタップも数は $4 \times X$ であり、以下同じである。

【0143】受信機は、同期等化器の訓練データのため、送信機と同じK記号シーケンスを使用する。等化器の長さは、記号シーケンス長さの倍数であるので、送信シーケンスと受信機基準シーケンスとの間の相対的位相は問題ではない。

【0144】一旦同期等化器の平均2乗誤差がしきい値の下に落ちると、セグメントAは検出済みである。受信

機は、適合プロセスを停止し、係数を分析する。それは、そこで係数を円のように回転し、大部分のエネルギーをもつN個の連続的係数が同期等化器の全面でグループを作る様にする。Nは、CAP復調に使用される直角適合フィルタの長さである（次の節を見よ）。これは、受信機の記号期間を送信機の記号期間と整列させる。

【0145】回転後、受信機は信号をフィルタし続けるが、同期等化器の係数は更新しない。同期等化器の出力は、そこで、長さKに整合したフィルタへ通過する。整合されたフィルタは、セグメントBの検出に使用される。このシーケンスは、2つの値しかもたないので、2進相関器も使用できる。

【0146】整合されたフィルタ（相関器）の出力が、しきい値より大きいと、受信機は、次の記号は訓練データの開始であることを知る。受信機は、今や、CAP復調に使用される直角適合フィルタを、実施する。これらは、僅かに間隔を置いた適合等化器であり、これらの長さは、実際の物理的チャネルのインパルス応答に依存する。これらの復調等化器は、セグメントCの公知の訓練データを使用して訓練される。訓練が終了した後、復調等化器は、判定で示されたモードに入り、ここでは、基フレーム番号

準データがCAPスライサから来る。

【0147】図34を参照すると、DMT回線コードと共に使用されるための、時間領域等化器訓練シーケンスが示される。DMTに対する、この発明の部分では、周波数領域訓練シーケンスの代わりに、図34に示す時間領域訓練シーケンスを使用する。訓練シーケンスの基本ユニットは、ランダムデータブロック $\{x_n\}$ 、 $0 \leq n < N$ である。全シーケンスは、ランダムデータブロック $\{x_n\}$ が、図34に示すように2ブロック毎に交番するデータブロックの符号と共に時間において反復する様に配列される。

【0148】記載を容易にする目的で、次のの表示法が使用される。時間領域等化器タップ w_1 、チャネルインパルス応答（時間領域等化器を含む） h_k 、等化器前の受信機データ $y_m[n]$ 、及び等化器後 $z_m[n]$ 、ここに m は、データブロック上のラベルを表す。図34における送信信号に対応する受信信号は次の数式6で示される。

【0149】

【数6】

$$1. \quad z_1[n] = \sum_{k=0}^n x_{n-k} \cdot h_k + \sum_{k=1}^n x_{N-k} \cdot h_{n-k} + p_n$$

$$2. \quad z_2[n] = -\sum_{k=0}^n x_{n-k} \cdot h_k + \sum_{k=1}^n x_{N-k} \cdot h_{n-k} + p_n$$

$$3. \quad z_3[n] = -\sum_{k=0}^n x_{n-k} \cdot h_k - \sum_{k=1}^n x_{N-k} \cdot h_{n-k} + p_n$$

$$4. \quad z_4[n] = \sum_{k=0}^n x_{n-k} \cdot h_k - \sum_{k=1}^n x_{N-k} \cdot h_{n-k} + p_n$$

$$5. \quad z_5[n] = z_1[n]$$

【0150】前の式において、 p_n は、訓練シーケンスに重ね合わされるパイロットトーンである。これら等式の右側の第2項は、前のフレームの記号間干渉に帰する。第2項は、演算を行うことにより分離できる。フレ

ーム4、-フレーム1、

【0151】

【数7】

$$err[n] = z_4[n] - z_1[n] = -2 \left(\sum_{k=1}^n x_{N-k} \cdot h_{n-k} \right)$$

【0171】

フレーム

【数16】

$$3. \quad z_1[n] = - \sum_{k=0}^n x_{n-k} \cdot h_k - \sum_{k=1}^n x_{N-k} \cdot h_{n-k} + p_n$$

$$5. \quad z_2[n] = - \sum_{k=0}^n x_{n-k} \cdot h_k + \sum_{k=1}^n x_{N-k} \cdot h_{n-k} + p_n$$

【0172】この、場合検出信号は、次式で表される。

【数17】

【0173】

$$\text{det}[n] = z_2[n] + z_1[n] = -2 \cdot \sum_{k=0}^n x_{n-k} \cdot h_k + 2 \cdot p_n$$

【0174】この検出フレームのパワーは、データフレームのものより、大きい、 $\text{pwr-det} > \text{pwr}$ 。一旦 $\text{pwr-det} > \text{pwr}$ が検出されると、受信データストリームにおいて、DMT受信機は、それが訓練シーケンスの終わりであることを決定する。訓練シーケンスの終わりに対するデータパターンは、フレーム5に挿入されているので、これは、時間領域等化器の訓練よりも、フレーム境界の検出に使用され、時間領域等化器の更新には影響しない。

【0175】時間領域等化器の訓練に続いて、送信機は、周波数領域等化器を訓練するため、別のシーケンス $\{y_n\}$ を送るべきである。周波数領域等化器訓練シーケンスは、正確に反復可能ブロック $\{y_n\}$ で作ることができる。図37は、全訓練シーケンスを示す。訓練シーケンス $\{y_n\}$ の体制において、 pwr-det は「高」のままである。

【0176】MDSLの回線管理部は、ホストソフトウェアが、MDSLを単一リンクモードをもつ専用回線の下で動作するように予備構成することを可能にしている。現在、MDSLは次のモードを使用する。

- 単一リンクをもつ専用回線 (LLSL)
- 多重リンクをもつ専用回線 (LLML)
- ソフトダイヤルをもつ交換回線 (SLSD)
- ハードダイヤルをもつ交換回線 (SLHD)

LLSLモードの下では、電気通信回線は、遠隔MDSLシステムをもつMDSL通信を専ら行う。この回線接続モードの下では、1つのデータリンクだけ許される。

【0177】LLMLモードは、同じ専用回線内で異なる速度での多重リンク接続を許すことを除いて、LLSLと同じ動作をする。リンクの数及び回線速度は、回線速度の容量に対して動的に構成できる。このモードの下では各リンクは、独立の専用回線の様に動作し、リンクに基づき事を除き同じ管理方式に従う。

【0178】SKSDモードは、交換MDSL回線上で動作し、この回線上でMDSL-Rは遠隔サーバにより制御されるMDSL-Cにより自動的にダイヤルされる。このモードの下では、回線管理は、普通の古い電話サービスとは関係の無い特別のMDSLダイヤルアップ手順に従う。このMDSLダイヤルアップ手順は、MDSLモデム内部初期化手順により定義される。それは2つのダイヤルアップIDをもち、1つはMDSL-Cポートに関係し、他方は、MDSL-Rモデムに関する、MDSL-CポートのためのIDは、加入者電話番号に加え1個の数字で、これは0として選ばれ、またMDSL-RモデムのためのIDは、加入者電話番号に加え1個の数字1に選ばれるものでよい。他の8個の値、2から9は保留される。

【0179】SLHDモードは音声帯域モデムに類似の方法で動作するが、MDSLダイヤルアップ手順を持つ。MDSLモデムは、電話番号を記憶するか、又はアプリケーションにより手動でダイヤルされる。

【0180】次に、モード動作の1例として、単一リンクモードをもつ専用回線の下でのMDSL回線接続管理を説明する。MDSL回線管理ホストインタフェースは、ホストソフトウェアが、データパケットを送信/受信する用意のある回線を構成できる様にする。ホストソフトウェアはまた、データの流れを停止するため、回線接続を手動で停止できる。

【0181】MDSL回線接続管理ホストインタフェースにおける、回線接続コマンドは、ホストインタフェースが回線を、MDSLの支持する回線モードの1つへ構成するためホストインタフェースに対して使用される。LLSLモードのもとでは、それは、また送信/受信データ速度、最大フレームの大きさ及びデータリンクプロトコルをセットアップする。このコマンドは、通常MDSL初期化中又は誤り回復プロセス中に呼び出される。このコマンドの実行が成功した後、構成下のMDSLは、

リンクを通してデータパケットを送信／受信する準備ができています。LLMLに対して、データリンクは、データの流れを許すためオープンされ／作成されている。NDLの回線構成は、非同期手順である。ホストは、MDSLにより発生する「回線接続済み」割り込みにより回線は旨く構成されたことが通知される。MDSLにおける回線構成プロセスは、図38に示される。

【0182】ホストインタフェース

MdsLineConfigure (IN Line Mode, IN TxSpeed, IN RxSpeed, TN MaxTxFrameSize, IN MaxRxFrameSize, IN TxProtocol)

回線コード (LineMod) 入力パラメタは、どの回線コードにMDSLが構成されるべきかを特定する。それは、次の定義をもつ。

- 0-単一リンクをもつ専用回線
- 1-多重リンクをもつ専用回線
- 2-ソフトダイヤルをもつ交換回線
- 3-ハードダイヤルをもつ交換回線

TxSeed及びRxSpeedは、上流および下流回線速度を与える。MaxTxFrameSize及びMaxRxFrameSizeパラメタは、データを送信及び受信するための最大フレームを特定する。TxProtocol 及びRxProtocolは、データの伝送のため使用される物理的層フレーミングプロトコルを特定する。現在、それは次の定義を持つ。

ビット0-ビット1フレーミングプロトコル名称

- 00-生 (Raw) MDSL (データパケット化なし)
- 01-MDSL特定パケット化
- 10-HDLC (高レベルデータリンク制御)
- ビット2 パケットヘッダの圧縮あるかどうかを示す
- ビット3 パケットデータの圧縮あるかどうかを示す
- ビット4 データ暗号化されているかを示す

【0183】MDSL回線接続管理ホストインタフェースにおいて、回線停止コマンド (HaltLineCommand) は、MDSLに、データ流れ制御のため、データの送信／受信を停止するよう知らせる。それは、内部データ送信バッファ及び状態フラグの全部を励起し、そして要求を通知するメッセージを遠隔MDSLに送り、そして手動で回線を「回線接続」状態にする。このコマンドは、回線が「回線接続」状態にあるときだけ効果を収める。その他のときは、誤りを戻す。回線停止は、非同期プロセスであり、ホストは、図39に示す様に、何時その回線が「回線切断」状態になったかの通知を受ける。

ホストインタフェース

MdsHaltLine ()

【0184】MDSLの内部には、回線ハンギング (hanging) 及び予期しない出来事を報告するために

回線状態の監視に使用する回線状態エンジンがある。MDSL専用回線モードにおいて次の回線状態が定義される。

- 回線電圧降下 (drop) - 回線のプラグが入れてないか破損し、物理的信号は受けられない。
- 回線切断 (disconnected) - 回線は物理的に接続されているが、データ伝送の用意ができていない。

- 回線接続 - 回線はデータパケットの送信／受信の用意ができています。MDSL回線接続管理ホストインタフェースは、回線状態情報を得るため2つの方法を与える。1つの方法は、回線状態入手 (Get Line Status) コマンドである。

MdsGETLineStatus (OUT Line STATUS, OUT LineConfigure) LineStatusパラメタは、MDSL回線状態情報を戻す。LineConfigureは、MdsLineConfigure () コマンドによりセットされた回線構成情報を記憶するのに使用される構造体。他の方法であつて、ホストソフトウェアが回線状態変化を通知される方法は、回線管理事象の登録によつてである。MDSLは、ある事象が起きたときホストソフトウェアが割り込まれることを許す。回線管理に関係ある事象は、次のものである。

- 回線接続：回線接続が、丁度確立された。
- 回線切断：前に接続された回線が、MdsHaltLine () 呼び又は予期しない事件で切断されている。
- 回線電圧降下：回線は物理的に切断されている。信号はこの回線では受けられない。図40に示すように、ホストソフトウェアが回線状態を周期的に調べることが可能のようにMDSLによりタイム割り込みが提供される。回線接続メッセージを、相互に接続された2つのMDSL端の間で交換する必要がある。これらのメッセージは、MDSLにおいて、特別回線管理パケットとして定義されている。MDSL-CとMDSL-Rの間で回線管理情報を交換するため次の種類の回線管理パケットが定義されている。

- 回線構成コマンドパケット
- 回線停止コマンドパケット
- 確認パケット

【0185】図41を参照すると、回線構成コマンドパケットの形式が示される。

- 回線モードは前に定義される
- データ送り出し速度
- データ受信速度
- 最大送り出しフレームサイズ
- 最大受信フレームサイズ
- データ送り出しプロトコルは前に定義される
- データ受信プロトコルは前に定義される

検査合計は、標準TCP/IPアルゴリズムを用いて計算される。メッセージにおける全部の16ビット整数の1の補数(検査合計フィールドを含む)

【0186】図42を参照すると、回線停止コマンドパケットの形式が示される。IDは、1オクテットでコマンドと解答の整合の助けとなる

長さは、オクテットでのパケット長さである

検査合計は、標準TCP/IPアルゴリズムを用いて計算される。メッセージにおける全部の16ビット整数の1の補数(検査合計フィールドを含む)

【0187】図43を参照すると、確認パケットの形式が示される。コードは、どの種類の確認パケットであるかを定義する。それは、次の定義を含む。

2-回線構成確認

4-回線構成拒否

6-回線停止確認

IDは、1オクテットでコマンドと解答の整合の助けとなる

長さは、バイトでのパケット長さである

状態コードは、次の定義をもつ

- 成功
- 未承認パケットID
- 構成データの一部受入不能
- 構成は完全に拒否
- 検査合計誤り

データは、0又は偶数のオクテットを含み、構成データのどの部分が遠隔端で受入不能かを特定する。検査合計は、標準TCP/IPアルゴリズムを用いて計算される。メッセージにおける全部の16ビット整数の1の補数(検査合計フィールドを含む)

【0188】電力投入後、MDSLは、その内部初期化プロセスを進める。このプロセスは、4つのステップを含み、チャンネル調査、回線コード選択、速度交渉及びトランシーバ訓練である。初期化手順の後、MDSL-Rは、スタンバイモードへ移る。この瞬間での回線状態は、前に定義した様に「切断」である。回線が物理的に接続されたことを検出するとホストソフトウェアは、MDSLLineConfirure()コマンドを、回線構成のためMDSL-Rへ送る。MDSL-Rは、そこで回線構成コマンドパケットを、構成データと共のMDSL-Cへ送る。回線構成コマンドを受信し、構成データを調べた後、MDSL-Cは、回線構成を確認するため確認パケットを送り出す。もしMDSL-Cが、構成データを受入不能であれば、それは、構成拒否パケットを送る。それは、またどの種類の誤りであるかを特定する状態メッセージを与える。もし、構成データの一部だけが受入不能であれば、データフィールドは、図44に示す様に、受入不能の構成データを含む。

【0189】接続が確立された後、それは、次の事象が起こるまで接続のままである。

- 回線はプラグが抜かれ、又は破損している

- MDSL-Rは電力低下である

- MDSL-Cはサービスしていない

MDSL-Cが、動作停止になろうとしているか、又はMDSL-Rが、電力低下すると何時でも、回線停止コマンドパケットが送り出される。コマンド送り元は、確認パケットが受信されるか、又は回線停止コマンドがタイムアウトするまでは送り続ける。受信側においては、回線が切断されていることを確認するため、確認パケットをメッセージ送り元へ送つた後、それは、全部の内部データバッファ及び状態フラグをクリアアップする。回線状態は、そこで「回線切断」状態に変わる。図45は、MDSL-Cが、それがサービスを止める前に、回線停止コマンドを送る例を示す。

【0190】MDSLホストは、簡単な、ユーザに順応した、効率的で低費用のインタフェースを、16ビットコントローラに提供することを意図している。ホストインタフェースは、次の機能を提供する。

- ホストとMDSLネットワークインタフェースカード(NIC)の間のコマンド/制御通信

- 回線接続管理

- 送信/受信データパケット

ホストコマンド/制御通信機能は、サービスを初期化する、もし、それがEEPROM内になればDSPコードをローカルRAMへダウンロードする、コマンドをMDSLへ送る、状態変化を監視及び報告する。

【0191】2つのMDSL-RとMDSL-Cの間の接続シグナリングは、異なる回線モード、ダイヤルアップ回線モード及び専用回線モードに従って違ってもよい。ダイヤルアップ回線モードは、基本的電話、-POTSに対応する保証された一組の機能を与える。このモードの下では、システムソフトウェア及びハードウェアは、MDSL-R位置の電話応用プログラミングインタフェース(TAPI)及びMDSL-C位置の電話サービスプロバイダインタフェース(TSPI)と接続を確立するため作業しなければならぬ。専用回線モードの下では、接続は、MDSLの初期化の直後に確立される。しかし、それは標準POTSサービスを提供しない。

【0192】物理的層のパケット化が、MDSL(HDLCのような)に対して使用するのが好ましい。PPPに対する最大パケットサイズは、1500バイトであるが、フレームに対して32バイトのオーバーヘッドを許すべきである。MDSLは、データをパケットバッファから回線へ送り、ホストに、パケットを既に送り出したことを通知する。それはまた、新しいパケットが受信バッファに入れられたときホストに通知する。送信及び受信バッファは、MDSL上の共用されるメモリでもよい。

【0193】次のコマンド及び制御が採用される。

1. リセット(Reset)

構文: Reset ()

記述: システムにおける全部のコマンド実行を停止する。転送/受信バッファを励起し、そして内部リセットを行う。

パラメタ: なし

戻り: なし

2. DSPモジュールにロード (Load DSP Module)

構文: LoadDspModule (ModuleAddr, ModuleSize)

記述: DSPモジュールをMDSLにロードする

パラメタ: ModuleAddr-DSDPモジュールアドレス開始

ModuleSize-DSDPモジュールサイズ

戻り: なし

3. 割り込みマスクセット (Set Interrupt Mask)

構文: SetInterruptMask (EventMask)

記述: 選択された事象 (s) に基づき、ホスト処理装置の割り込みを可能にする。パラメタ: EventMaskは、割り込みマスクに対して16ビット整数値である。

【0194】表5は、マスクにおけるビットを識別する。ビットに対する1の値は、このビットに対応する割り込みを可能にする。表中で定義していない全部のビットは将来のため保留され、ゼロにセットすべきである。

【0195】

【表5】

ビット #	Mnemonic	事象	事象定義
0	INTE	割り込み可能	このビットは、EventMaskにより割り込みセットを可能にする。
1	LNC	回線接続	回線接続が確立された。
2	LMDC	回線切断	前に接続された回線が切断された。
3	BFOVF	Rxバッファオーバーラン	回線受信バッファはオーバーランしている。
7	BFEHP	Txバッファエラー	回線転送バッファはエラーである。
9	PKARV	パケット到着	パケットは回線受信バッファに入力された。
10	PKST	パケット送り	パケットはディマ回線送信バッファから取り出された。
15	TINT	タイマ時間切れ	現在のタイマカウンタは0になる。

a. もし送信バッファが一度に1個のバッファを保持できれば、この割り込みは割り込みに送られたパケットについて冗長である。

【0196】4. 割り込み状態入手 (Get Interrupt Status)

構文: GetInterruptStatus ()

記述: 選択された事象 (s) の発生に基づき、割り込み状態を入手

パラメタ: なし

戻り: EventStatus. MDSLは、表5のEventMaskパラメタの定義に対応する16ビット状態数を返す。この機能の呼び出しは、丁度始まった割り込みをクリアする。

【0197】次の回線接続管理コマンドが利用できる。

1. 回線構成 (Line Configuration)

n)

構文: LineConfigure (LineMode)

記述: データパケットの受信/送信を用意するため回線を構成する。

パラメタ: LineModeは、どの種類の回線モードMDSLを構成すべきかを指示する。それは次の表に示すビット定義をもつ。定義されない全ビットは将来の使用のため保留される。

【0198】

【表6】

ビット# Mnemonic	事象	事象定義
0	回線リセット	このビットがセットされると、データ回線は専用回線モードに動作する。このビットがクリアされると、データ回線はダイヤルアップモードに動作する。
1	音声回線フラグ	このビットがセットされると、音声信号を送信し、同時に MDSL に与えるデータ信号送信に機能する。
2~6	速度定義	これらのビットは、送信及び受信の速度を定義する。ビット 6 は、速度が送信又は受信のために示す。ビット 2 から 5 は、16 の異なる速度を示す。

戻り: なし

【0199】2. 回線状態入手 (Get Line Status)

構文: GetLineStatus ()

記述: このコマンドは、現在の回線状態を指示するため、16 ビット数を返す。

パラメタ: なし

戻り: LineStatus. この戻される数の定義は、次の表 7 にある。

【0200】

【表 7】

ビット# Mnemonic	事象	事象定義
0-1	回線状態	これらの 2 つのビットは次の回線状態を示す。 1. 回線がオン (物理的信号は受信された) 2. 回線が切断 (送受信の準備なし) 3. 回線が接続 (データ送受信の準備あり)
2	回線モード	このビットがセットされると、回線は専用回線モードに動作し、このビットがクリアされると、ダイヤルアップモードに動作する。
3	音声回線フラグ	このビットがセットされると、音声信号を送信し、同時に MDSL に与えるデータ信号送信に機能する。このビットがクリアされると、音声帯域は同時に機能しない。
4-8	速度定義	これらのビットは、送信及び受信の速度を定義する。ビット 8 は、送信又は受信の速度を示す。ビット 4 から 7 は、16 の異なる速度を定義する。

【0201】3. 回線接続停止 (Halt a Connected Line)

構文: HaltLine ()

記述: データ流れの制御のためデータの送信/受信の停止を MDSL に告げる。

それは内部バッファの全部を励起し、回線を手動で「切断」状態にする。

パラメタ: なし

戻り: なし

【0202】次の送信/受信データパケットコマンドが利用できる。

1. パケット送り (Send Packet)

構文: SendPacket (DataPtr, Size)

記述: このコマンドは、1 つのデータパケットが MDSL 送信バッファにコピーされたことを MDSL に告げる。パケットがバッファから、移動された後に割り込みが発生するであろう。

パラメタ: DataPtr は、データパケットが記憶されている送信バッファのメモリアドレスを示す。パケットの長さは、最大の許可パケットサイズより小さいか等しい。

戻り: なし

2. 受信情報検査 (Check Receive In

formation)

構文: CheckReceiveInfo (DataPtr, Size, ErrorFlag)

記述: このコマンドは、受信バッファ中にパケットがあるかどうかにより、真 (1) 又は偽 (0) を返す。

パラメタ: DataPtr が、パケットが記憶されているメモリアドレスを戻すため使用される。Size が、受信パケットのサイズを戻すため使用される。もし、伝送中に誤りが発生すると、ErrorFlag は、非ゼロにセットされる。

戻り: 1-受信バッファ中にデータがある。

0-受信バッファ中にデータが無い。

3. 送信情報検査 (Check Sending Information)

構文: CheckSendInfo ()

記述: このコマンドは、MDSL 送信バッファが空のとき 0 を返す。その他の場合は、バッファに残されたバイトの数を返す。

パラメタ: なし

戻り: 送信バッファデータサイズ。それは、送信バッファが空のとき 0 である。その他の場合は、バッファに残されたバイトの数を返す。

【0203】図46は、ウィンドウズ95又はウィンドウズNT環境をもつホストと共に使用されるドライバのソフトウェア構造を示すが、これは、住宅のパーソナル計算機における普通の場合である。図47は、ソフトウェアドライバ構造をより一般的に示す。MDSL NICのシステムソフトウェアは、ウィンドウズNT3.5及びウィンドウズ95オペレーティングシステムのためのNDIS3.0WANミニポートとして実施されてきた。次のものは、次の3つの見地から分析したものである。

1. ウィンドウズNT3.5及びウィンドウズ95の下で、ミニポートデバイスドライバとして実施されるシステムソフトウェアによるシステムの機能性。
2. システムが行う入力及び出力データ処理。
3. システムソフトウェアとNDISライブラリとの相互作用。//

【0204】MDSLドライバは、ネットワークのミドルアクセスコントロール (MAC) サブレイヤを制御し、管理するためNDISミッドドライバとして実施される。その構造は図46に示される。それは、ウィンドウズNT3.5又はウィンドウズ95インターネットシステムソフトウェア内のコンポーネントである。MDSLドライバは、NDIS3.0に規定されたインタフェース及びデータ構造の定義に従う。ドライバは、それを機能させるため、システムの中に設置され又は集積される必要がある。

【0205】MDSLドライバは、WANネットワークインタフェースカードドライバとして機能する。それ

は、上部縁上でプロトコルドライバと相互作用し、そして下部縁上でMDSL NICを制御する。これらの全部の相互作用及び制御は、ウィンドウズNT/ウィンドウズ95内のNDISライブラリ又はNDISラッパを通る。

【0206】図48は、システムソフトウェアにおけるデータ流れの通路を示す。図48は、どのようにして受信データがNICカードにより受信され、そしてドライバに渡され、そこで種々の機能を經由して輸送インタフェースへ渡され、そして、どのようにしてドライバに戻されるかを示す。MDSLドライバはMDSLネットワークアダプタと共に現れる。それは、MDSL-Rと一緒に家庭のPC内に設置され、そして同じドライバで走行するMDSL-Cに接続され、この場合MDSL-Cの変調アルゴリズムは異なることもある。インターネットルータが、MDSL-Cの位置にあると、MDSL-Rは、TELNET、FTP及びNetScapeの様な多くのインターネットをMDSL NICを通して走行させることができる。データ通信と音声通信は同時に生じる。

【0207】次のエントリ点及び機能は、完全にNDIS3.0仕様書に従っている。ドライバエントリ点 (ドライバエントリ) (Driver Entry) は、ドライバがメモリの中にロードされた時、オペレーティングシステムにより、呼びだされる主要なエントリ点である。

入力

DriverObject: オペレーティングシステムにより作られた、ドライバオブジェクトへのポインタ
RegistryPath: レジストリパラメタを読み取りに用いるレジストリパスへのポインタ

出力

Return Value: STATUS-SUCCESS or STATUS-UNSUCCESSFUL
DriverEntryは、次のことを行う。

1. NDIS WANラッパを初期化するためNdisMInitializeWrapperを呼び出す。
2. 特性表を初期化し、MDSLドライバのエントリ点をNDIS WANラッパへ輸出する。
3. MDSLドライバをNDIS WANラッパへレジスタするためNdisMRegisterMiniportを呼び出す。図49は、ドライバエントリのためOS、NDISライブラリとMDSLドライバの間の相互作用を示す。

【0208】初期化エントリ点 (MdsIInitialize) は、MDSLモデムを初期化するため、NDISライブラリにより呼び出される。

入力

MediumArray: NDISライブラリにより支持される全ネットワークメディア

MediumArraySize:メディア配列における要素の数

MdslAdapterHandle:NDISライブラリにより割り当てられるMDSLドライバを識別するハンドル

NdisConfigContext:NDIS構成のためのハンドル

出力

OpenErrorStatus:もし戻り値がNDIS-STATUS-OPEN-ERRORであれば、MDSLドライバは、このパラメタを、誤りについての情報を特定する状態値にセットする。

SelectedMeriumIndex:MDSLドライバは、この指標を、MDSLドライバのメディア型を特定するMediumArrayにセットする。

戻り値:MdslInitializeは、NDIS-STATUS-SUCCESSを返し、又はそれは、次の状態値に戻すことができる。

NDIS-STATUS-ADAPTER-NOT-FOUND

NDIS-STATUS-FAILURE

NDIS-STATUS-NOT-ACCEPTED

NDIS-STATUS-OPEN-ERROR

NDIS-STATUS-RESOURCESS

NDIS-STATUS-UNSUPPORTED-MEDIA

処理

MdslInitializeは、次のことを行う。

1. そのメディア整合を見出だすため、MediumArrayを通して探索する。もし、整合が見付からないと、NDIS-STATUS-UNSUPPORTED-MEDIAは戻される。
2. MDSL NIC (割り込み番号、ボード名称、チャンネルアドレス又は回線アドレス、スイッチ形式等) 全構成情報を得る。
3. MDSLのためのメモリ、ドライバデータ構造の割当及び初期化。
4. MDSL NICをもつ関連するMdslAdapterHandleを含むMDSL NICの物理的属性をNDISラツパに知らせる。
5. MDSL NICの物理的位置をシステムアドレススペースにマップする。
6. MDSL NICをリセット又は初期化する。
7. 送信キューをセットアップ及び初期化する。
8. 割り込みを初期化する。
9. 回線を初期化する。図50は、NDISライブラリとMdslInitializeのためのドライバの間の相互作用を示す。

【0209】エントリ点は、MDSL NICにハードウェアリセットを発行し、またそのソフトウェア状態を

リセットする。

入力

MdskAdapterContext:MiniportInitializeにより初期化されたハンドル。

出力

AdressingReset:もし、NDISライブラリが、アドレス情報を現在の値に回復するためMisslInformationを呼び出す必要があれば、真(TRUE)にセットする。

戻し値:なし

処理

MdslResetはMDSL NIC上にソフトウェアリセットを発行する。それは、またMDSL NICのパラメタをリセットする。もし、MDSLNICのハードウェアリセットが現在の位置アドレスをリセットすれば、MDSLドライバは、リセットに続いて、自動的に現在の位置アドレスを回復する。図51は、NDISライブラリとMdslResetのためのドライバの間の相互作用を示す。

【0210】エントリ点(MdslReconfigure)は、MDSL NICを、NDISライブラリ機能において利用できる新しいパラメタへ再構成するためNDISライブラリにより、呼びだされる。それは、プラグ及びプレイアダプタ及びソフトウェア構成可能アダプタを支持するため使用され、これらは、走行時間中に変更されるパラメタをもつてもよい。

入力

MdslAdapterContext:MiniportInitializeにより初期化されるハンドル。

WrapperConfigurationContext:NDIS構成のハンドル。

出力

OpenErrorStatus:このパラメタは、もし戻り値がNDIS-STATUS-OPEN-ERRORであれば、誤りについての情報を特定するためMDSLドライバによりセットされる。

戻り値:

NDIS-STATUS-SUCCESS

NDIS-STATUS-NOT-ACCEPTED

NDIS-STATUS-OPEN-ERROR

処理

NDIS-STATUS-NOT-ACCEPTEDを返す。

【0211】エントリ点(MdslHalt)は、MDSL NICを停止するためNDISライブラリにより呼びだされる。

入力

MdslAdapterContext:MdslIn

initializeにより初期化されるハンドル。

出力

なし

処理

MdsslHaltは、次のことを行う。

1. 割り込みハンドリングのレジスタを解除する。
2. システムからのメモリのマップを止める。
3. システムメモリを解除する。

図52は、NDISライブラリとMdsslHaltのためのドライバの間の相互作用を示す。

【0212】エントリ点(MdsslCheckForHang)は、MDSL NICの状態を定期的に調べるためNDISライブラリにより呼びだされる。

入力

MdsslAdapterContext:MdsslInitializeにより初期化されるハンドル。

出力

戻り値: もしMDSL NICが動作していなければ真(TRUE)。

処理

MDSL NICの状態を調べる。

【0213】エントリ点(MdsslEnableInterrupt)は、MDSL NICが割り込みを発生できる様に、NDISライブラリにより呼びだされる。

入力

MdsslAdapterContext:MdsslInitializeにより初期化されるハンドル。

出力

戻り値: なし

処理

MDSL NICハードウェアの割り込みを発生を可能にする。

【0214】エントリ点(MdsslEnableInterrupt)は、MDSL NICがどのような割り込みも発生できない様に、NDISライブラリにより呼びだされる。

入力

MdsslAdapterContext:MdsslInitializeにより初期化されるハンドル。

出力

戻り値: なし

処理

MDSL NICハードウェアの割り込みを発生を不可能にする。

【0215】MdsslISRは、MDSLの割り込みサービスルーチンのエントリ点。

入力

MdsslAdapterContext:MdsslInitializeにより初期化されるハンドル。

出力

InterruptRecognized: もし、MDSL NICが、割り込み回線を共有し、また、割り込みが、そのNICからきたことを検出すれば、MDSLドライバは、そのパラメタを真(TRUE)にセットする。

QueueMdsslHandleInterrupt: もし、MDSL NICが、割り込み回線を共有し、また、もし割り込みのハンドリングを終えるためMdsslHandleInterruptが呼びだされなければならないときは、このパラメタを真(TRUE)にセットされる。

戻り値: なし

処理

この機能は、割り込みに応答して高い優先度で走行する。MdsslHandleInterruptは、より低い優先度に残す。それは、次のことを行う。

1. 割り込みの理由を入手
2. ハードウェアにおける割り込みをクリアする。
3. InterruptRecognized及びQueueMdsslHandleInterruptに従ってセットする。

【0216】エントリ点(MdsslHandleInterrupt)は、割り込みを処理するためNDISライブラリにおける延期された処理ルーチンにより呼びだされる。

入力

MdsslAdapterContext:MdsslInitializeにより初期化されるハンドル。

出力

戻り値: なし

処理

MdsslHandleInterruptは、次のことを行う。

1. 割り込みの理由を得るためMDSL-NICを調べる。
2. 次の可能な割り込みを1つずつ処理する。
 - パケットは丁度受信バッファの中に入れられた
 - パケットは丁度送り出された
 - 回線は丁度接続された
 - 回線は切断された
 - 回線は電圧降下した
 - 受信バッファオーバーラン

【0217】エントリ点(MdsslQueryInformation)は、MDSLドライバの能力及び状態を質問するためNDISライブラリにより呼びだされる。

入力

MdsslAdapterContext:MdsslInitializeにより初期化されるハンドル。

OID: ドライバが動的構成情報及び統計的情報を蓄積

する管理情報ブロック中の管理オブジェクト（又は情報要素）のオブジェクトID。その形式及び定義についてNDIS3.0仕様書を参照。

InformationBuffer: 情報を受けるバッファ

InformationBufferLength: InformationBufferのバイトでの長さ
出力

ByteWritten: InformationBufferに実際に書き込まれたバイトの数

BytesNeeded: 特定されたオブジェクトの完全な情報を得るため必要な追加のバイトの数。

戻り値: Mds1QueryInformationは、NDIS-STATUS-SUCCESS又は次の状態値を戻す。

NDIS-STATUS-INVALID-DATA
NDIS-STATUS-INVALID-LENGTH

NDIS-STATUS-INVALID-OID
NDIS-STATUS-NOT-ACCEPTED
NDIS-STATUS-NOT-SUPPORTED
NDIS-STATUS-PENDING
NDIS-STATUS-RESOURCES

処理

MDSLドライバは、次のOIDを同期して確認するだけである。

OID-GEN-HARDWARE-STATUS: MDSL NICのハードウェア状態を調べる。

OID-GEN-MEDIA-SUPPORTED: Mds1MediumWanを戻す。

OID-GEN-MEDIA-INUSE: Mds1MediumWanを戻す。

OID-GEN-MAXIMUM-LOOKAHEAD: 最大パケットサイズ(1532バイト)を戻す。

OID-GEN-MAXIMUM-FRAME-SIZE: MDSLに対する最大フレームサイズを戻す(1500バイト)。

OID-GEN-LINK-SPEED: MDSLのリンク速度を戻す(384000bps)。

OID-GEN-TRANSMIT-BUFFER-SPACE: 最大パケットサイズ(送信バッファには1つだけのパケットが許されると仮定)を戻す。

OID-GEN-RECEIVE-BUFFER-SPACE: 受信バッファにおける最大パケットサイズ(1つだけのパケットが許されると仮定)を戻す。

OID-GEN-TRANSMIT-BLOCK-SIZE: 最大パケットサイズを戻す。

OID-GEN-RECEIVE-BLOCK-SIZE: 最大パケットサイズを戻す。

OID-GEN-VENDOR-ID: vendor

IDを戻す。

OID-GEN-VENDOR-DESCRIPTION: vendor記述ストリングを戻す。

OID-GEN-CURRENT-LOOKAHEAD: 最大パケットサイズを戻す。

OID-GEN-MAC-OPTIONS: 次のビットをセットする。NDIS-MAC-OPTION-RECEIVE-SERIALIZED、NDIS-MAC-OPTION-NO-LOOPBACK及びNDIS-MAC-OPTION-TRANSFERS-NOT-PEND

OID-GEN-DRIVER-VERSION: MDSLドライバの大及び小バージョン番号を戻す。

OID-GEN-MAXIMUM-TOTAL-SIZE: 最大パケットサイズを戻す。

OID-WAN-MEDIUM-SUBTYPE: MDSLは夫だMicrosoftにより定義されていないので、NdisWanIsdnが戻される。

OID-WAN-GET-INFO: NDIS WAN情報構造を戻す。

OID-WAN-PERMANENT-ADDRESS: WANアドレスを戻す。

OID-WAN-CURRENT-ADDRESS: WANアドレスを戻す。

OID-WAN-GET-LINK-INFO: Mds1LinkContextを戻す。他の全部のOidに対しては、NDIS-STATUS-INVALID-OIDを戻す。図53は、NDISライブラリとMds1QueryInformationのためのドライバとの相互作用を示す。

[0218] エントリー点 (Mds1SetInformation) は、MDSLドライバにより維持されている情報を変更するためNDISライブラリにより呼びだされる。

入力

Mds1AdapterContext: Mds1Initializeにより初期化されるアンドル。

OID: ドライバが動的構成情報及び統計的情報を蓄積する管理情報ブロック中の管理オブジェクト（又は情報要素）のオブジェクトID。その形式及び定義についてNDIS3.0仕様書を参照。

InformationBuffer: 情報を受けるバッファ

InformationBufferLength: InformationBufferのバイトでの長さ
出力

ByteRead: InformationBufferから読みだされるバイトの数

BytesNeeded: OIDを満足させるのに必要な追加のバイトの数。

戻り値: Mds1QueryInformation
は、NDIS-STATUS-SUCCESS又は次の
状態値を戻す。

NDIS-STATUS-INVALID-DATA
NDIS-STATUS-INVALID-LENGTH
NDIS-STATUS-INVALID-OID
NDIS-STATUS-NOT-ACCEPTED
NDIS-STATUS-NOT-SUPPORTED
NDIS-STATUS-PENDING
NDIS-STATUS-RESOURCES

処理

MDSLドライバは、次のOIDを同期して確認するだけである。

OID-GEN-CURRENTY-LOOKAHEAD: NDIS-STATUS-SUCCESSを直接な
にもしないで戻すが、それは、WANドライバは常に、
LOOKAHEADサイズに無関係に全パケットを示す
からである。

OID-GEN-WAN-SET-LINK: InformationBufferに記憶されているMds1
LinkContextをMDSL WanLinkIn
fo構造の中へコピーする。他の全部のOIDに対し
ては、NDIS-STATUS-INVALIDを戻
す。図54は、NDISライブラリとMds1SetI
nformationのためのデバイスとの間の相互作
用を示す。

【0219】機能 (Mds1ReceivePacket)
は、パケット受信割り込みのハンドリングのためM
ds1HandkeInterruptにより呼びださ
れる。この機能はNDIS Mds1Transfer
Dataを置換するため使用されるが、それはMDSL
ドライバは、データを受信バッファからプロトコルスタ
ックへ転送するためMds1TransferData
を呼びださないからである。

入力

Mds1AdapterContext: MDSLアダ
プタはMds1Initializeにより初期化され
る

出力

なし

戻り値: なし

処理

Mds1ReceivePacketは、次を行う。

1. データ伝送中に誤りがあるかどうかを見るため受信
状態を調べる。悪いパケットを除外し、誤りをNDIS
ラッパに示す。NdisMWanIndicateRe
ceiveを呼びだし、パケットが到着し、全パケット
は検査に利用できることを示す。
3. もし上の呼び出しが、NDIS-STATUS-S

UCCESSを戻すならば、NdisWanIndic
ateReiveComplteを呼びだし、受信事象
の終わりを示す。図55は、NDISライブラリとMD
SLReceivePacketの相互作用を示す。

【0220】エントリ点 (Mds1WanSend)
は、NDISライブラリにより呼びだされ、パケットを
アダプタを通りメディアに送信するようMDSL N
ICドライバに命令する。もし、メディアがこの呼び
がきた瞬間、使用中であれば、NDISドライバは送る
コマンドをキューし、又は最大送信値を低くする。

入力

Mds1BindingHandke: ハンドルは、M
ds1Initializeから戻される。

Mds1LinkHandle: 回線が接続されると、
ハンドルはNDIS-MAC-LINK-UP表示から
戻される。

WanPacket: NDIS-WAN-PACKET
へのポインタは構成バッファへのポインタを含む。

出力

状態: もし、戻り値がNDIS-STATUS-SUC
CESS又はNDIS-STATUS-PENDING
でないとき、誤りについても情報を特定する状態値。

戻り値: Mds1WanSendは、NDIS-STA
TUS-SUCCESS又は次の状態値を戻す。NDI
S-STATUS-PENDING

NDIS-STATUS-FAILURE

処理

Mds1WanSendは、次を行う。

1. 有効かどうか、を確かめるため、パケットサイズ
を調べる。
2. 回線が現在接続されているか、を調べる。
3. もし、メディアが現在使用中でなければ、パケッ
トを直ぐ送り、そしてNDIS-STATUS-SUC
CESSを戻す。もし、使用中であれば、パケットを送
信リストに入れ、そしてNDIS-STATUS-PE
NDINGを戻す。このパケットが送り出された後、M
DSLドライバは、NdisWanSendCoTnp
leteを呼び出し、送り事象の完了を示す。図56
は、NDISライブラリとMds1WanSendのド
ライバの相互作用を示す。

【0221】システム統合

ウィンドウズNT又はウィンドウズ95の下で、種々の
ネットワークソフトウェアコンポーネントは一緒にリン
クされ、又は図57に示す様にローカル階層の中に拘束
されている。ネットワークコンポーネントが、設置され
ると、情報が、ウィンドウズNTレジストリに書き込ま
れ、これはネットワークコンポーネントがロードされる
順序、及びどの様にネットワークコンポーネントが一
緒に結合されるかを記述する。ウィンドウズ95制御パ
ネルネットワーク・アプレット (Applet) (NCP

A) は、ネットワークコンポーネント設置及び結合を管理する。ドライバ結合は、図58に示す様に動作する。

【0222】システムのための外部インタフェースは、次の通りである。

ユーザインタフェース

MDSLドライバは、端（エンド）ユーザには直接露出していない。それは、NDISラッパを介して、システム内のプロトコルスタックと結合される。アプリケーションは、それを、ウィンドウソケット、NetBIOS、RPC等を通じて使用する。

ハードウェアインタフェース

MDSLドライバのハードウェアインタフェースは、MDSL、ホストインタフェース要求仕様書に記載されている。

ソフトウェアインタフェース

MDSLドライバは、13個の上部縁（Upper Edge）機能及びオペレーティングシステムへの1個のドライバ主エントリ点を備える。それは`ndis.lib`及び`ndiswan.lib`に定義されている機能を呼び出して、特定のネットワークインタフェースカード（NIC）と無関係な多くのタスクを実施する。

通信インタフェース

受信及び送信されているパケットは、NDIS WANライブラリにより提供されるどの形式でもよい。それは、ヘッダ圧縮、マイクロソフト2地点間圧縮、暗号化をもち又はもたないIPデータグラム（gram）または他のフレームでよい。また、もし簡単なHDL CフレームングスイッチがNDIS WANライブラリ内でターンオンしていれば、簡単なHDL Cフレームでよい。全部のこれらの高い層のフレームングは、MDSLドライバに明白である。

設計制約

設計は、NDIS 3.0 WANドライバ仕様書に従わなければならない。

属性

利用可能性／回復

エントリ点処理中の誤りは、ドライバの破滅的故障にはならないであろう。誤りは、呼び側に渡され、そしてNDISが適切な処理をする。MDSL NICの初期化又は回線接続の確立における故障は、誤りが呼び側に戻されることにはならないであろう。パケットの受信／送信中の誤りは、記録される。

ソフトウェア／取得

DSL帯域動作についてのマルチモードモデムは、フラッシュEPROM（フラッシュEPROM含む様に強化されたDSLモデムのボードバージョンの図17を見よ）にダウンロードすることにより取得できる。このダウンロードは、既にマルチモードモデムの中にある音声帯域構成（V. 34）を使用することにより遂行できる。特に、ホストは、音声帯域モデム動作を電話番号の

呼び出しに使用することができ、ホストは、次にソフトウェアを、音声帯域を通してのDSL音声帯域動作をフラッシュEPROMへダウンロードすることができる。同じやり方で、DSL帯域ソフトウェアは、音声帯域を通して又はDSL帯域を通してダウンロードされることができる。

【0223】図60を参照すると、上流及び下流に対するMDSL周波数分割が判る。音声帯域モデムにおいて、関係ある最高周波数は、3.3 KHzに過ぎない。MDSLにおいて、関係ある最高周波数は、数百KHzとなり得る。例えば、1/4 T1速度に対して、上流チャンネル F_{C1} の中心周波数は100 KHzであり、他方、下流チャンネル F_{C2} の中心周波数は300 KHzである。各チャンネルの帯域は、200 KHzであり、関係ある最高周波数は、 $F_{2+}=400$ KHzである。データの処理を低価格のプログラマブルデジタル信号処理装置（DSP）で処理するよう挑戦することは可能である。この発明は、図60に示されるいずれの通過帯域信号もDSPに対して同一に見えるようにすることにより、どの様にして、処理要求事項を減少させるかに向けられている。

【0224】MDSLモデムは、2つのモードをもち、中央局（CO）モード及び遠隔ユーザ（RU）モードである。中央局（CO）モードにおいては、モデムは高い周波数帯域で送信し、低い周波数帯域で受信する。ユーザ（RU）モードでは、この逆が生じる。モデムは低い周波数帯域で送信し、高い周波数帯域で受信する。

【0225】ナイキスト（Nyquist）サンプリング理論の通常の解釈を用いると、関係する高い周波数の2倍の最少サンプリング速度が、データの処理に必要である。COモデムに対して、アナログデジタル変換器（ADC）は、受信信号を F_{1+} の2倍でサンプルできる。しかし、それは、サンプルを、デジタルアナログ変換器（DAC）に対して、 F_{2+} の2倍で発生しなければならない。RUモデムに対して、DACは F_{1+} の2倍で走行できる。しかし、ADCは、 F_{2+} の2倍で走行しなければならない。

【0226】この発明は、サンプリング速度を減少させ、そして、その結果としてMDSLモデムの実施のための処理要求事項を減少させるため、デジタルサンプリング速度変換を使用する。

【0227】RUモデムに対して、高いサンプリング速度は、アナログデジタル変換プロセスに関係する。1/4 T1の例としてのモデム受信機フロントエンド（front end）が、図61のRUモデムに示される。受信アナログ信号は、その中心は300 KHzであるが、関係の帯域幅を隔離して信号対雑音比を最大にするため、最初にバンドパスフィルタを通す。信号は、次にADCにより、 f_{2+} の2倍、800 KHzの通常のナイキスト速度でサンプルされる。

【0228】デジタル領域のサンプルされたスペクトルは図63に示される。 $F_{\text{sampling}}/4=200\text{ KHz}$ より低い信号はないので、サンプリング速度は、これらサンプルを2でデシメイトすることにより、 400 KHz へ安全に減少させることができる。2によるデシメーションは、図63に示す様に、 100 KHz が中心の反転された画像を発生する。

【0229】原画像は、デシメイトされたデータストリームを1つ置きに(-1)で掛け算することにより得られる。ADCからの1つ置きの出力は廃棄されるので、これらを発生する必要は無く、即ち、ADCは、 800 KHz の代わりに 400 KHz で走行できる。

【0230】COモデムに対しては、デジタルアナログ処理において、高い出力サンプリング速度が要求される。それは、高い通過帯域信号に対応する出力サンプルを直接発生するため最少限、 800 KHz のサンプリング速度が要求される。もし、COモデムが、低い周波数帯域で出力サンプルを発生することができ、そして、幾分自動的にスペクトルを高い帯域へ移行できるとすれば、遥かに良いことである。図64は、デジタル領域における低い帯域信号を示す。

【0231】移行は、より高い速度へデジタル的にアップサンプリングすることにより作られる別名(alias)の画像を使用することにより達成される。2から 800 KHz によるアップサンプリングは、ゼロ値サンプル(zero valued sample)を計算された出力サンプルの間に挿入することからなる。これは元の 400 KHz サンプリング周波数の調波で画像を発生する。新しい修正された出力データストリームがDACに渡されると、図65に示すアナログ出力スペクトルが発生する。(変換プロセスによるサイン(sine)ロールオフ特性は図から除いてある。)適当なアナログバンドパスフィルタの使用により、 300 KHz を中心とする反転画像が選択できる。挿入された値は、ゼロであるので、これらはDSPにより計算される必要はない。反転は、奇数サンプルを(-1)で掛け算するか、又は完全に廃棄することにより訂正できるが、何故なら、スペクトルは、RUモデムでのデシメーションプロセスにより再度反転されるからである。図66に示す様にゼロサンプルインターリービングプロセスが、DSPの外部の簡単な外部論理により実施できる。

【0232】結論として、サンプリング速度変換の適用は、MDSL内のDSPが、それが常に、低い周波数帯域においてのみ送信し、受信していると考えることを可能にしている、その計算は、従つて、通常実際の信号周波数内容で示されるよりも、遥かに低いサンプリング速度に基づいている。

【0233】離散的マルトーン(DNT)が、非対称デジタル加入者回線(ACSL)もための標準として、ANSI標準委員会T1E1, 4により選ばれてい

る。T1E1標準活動に対する前の貢献は、高速フーリエ変換FFTにおける19ビット精度が、ADSL-2ビット速度(6-7Mbps)に対して適切な動的範囲を達成するのに必要であることを要求した。問題は、固定点16ビット処理装置においてどの様にFFTを実施し、そしてADSL-2ビット速度に対して、適切な動的範囲を与えるかである。

【0234】通常、固定点(fixed point)FFTを、実施するとき、各段階において、データは盲目的にスケールダウンされ、乗算及び加算演算中に固定点値がオーバーフローするのを防止する。もし、データ値の範囲が、この段階中にオーバーフローが起こらない様なものであれば、ダウンスケールは、不必要な精度の損失を生じる。

【0235】本発明の、教示に従つたこの問題の解決方法は、好ましくは16ビット固定点における、順方向及び逆方向FFTの両方を、可変スケール方式を使用して実施することであり、この方式は、各FFT段階の前のデータを検査し、そしてもしこの段階中にオーバーフローの可能性があると時のみ、データをスケールダウンする。これは、オーバーフローが起きなくても盲目的にスケールダウンすることにより生じる不必要な精度の損失を除去する。スケールアップの必要性は、各段階の前のFFTデータにおける符号ビットの数を見て決定される。右へのシフトによりデータのスケールアップは行われる。1ビットだけ、又は1度に2ビットをシフトすることによりテストが行われた。シフトしてテストがおこなわれた。一般に、両方のシフト量は、作動したが、データ値が最大値でまた特定のサイン/コサイン値の場合には、単一のシフトされた値でもオーバーフローした。

【0236】FFT中のスケールアップの合計量は、FFTの完了のとき、FFT出力データが正規化(再スケールされた)できるように維持される。添付の付録(Appendix)は、解答をテストするため使用したCコードのテストバージョンである。可変スケール方法は、「盲目的」スケールアップよりも多くの電力を要しないが、それは全部のデータは、FFTの各段階前で検査されねばならないからである。シミュレーション結果は、19ビット固定点固定スケールFFTは、期待された信号対雑音動作範囲において、16ビット可変スケールされた固定点FFTよりも「ぎりぎり」にしか良くなかった。

【0237】固定点FFTの可変スケールアップは、どの段階上でもオーバーフローが起きないデータ範囲で、そして精度の改良のため追加電力が利用できるいかなるアプリケーションにおいても利点を提供する。

【0238】中央局端において、多重MDSL回線をハンドルするため、モデムプールが使用できる。フロントエンド及び回路を結合するその目的専用の線が、各MDSL回線に必要であるが、高性能DSPチップの信号処

理パワーは多重MDSL回線間で共用できる。MDSLモデムブールの多重回線能力は、さらに、単一モデムブールユニット内に多重DSPチップを組み入れることにより強化できる。

【0239】図67は、MDSLモデムブールは、N個の論理MDSLモデムをもつことができ、各々は送信機部及び受信機部からなるものを示している。モデムブールの場所に起因して、送信機は、同じ中央局クロックに同期することができる。MDSL回線集中及び共用モデムブールアーキテクチャの故に、送信信号のデータ記号及び受信信号のサンプルは、全論理モデム間で容易にアクセス可能である。送信信号同期及び送信信号及び受信信号のアクセス可能性は、NEXT打ち消し（キャンセル）技術の適合を可能にする。多重入力-多重出力NEXTキャンセルは、MDSLモデムブールと共に実施できる。

【0240】NEXT及びエコー打ち消しハードウェアの費用を避けるため、好ましいMDSLモデムは、中央局から加入者へ下流方向に送信し、これとは逆に上流方向に送信するため、周波数分割多重を使用する。下流送信は、通常、MDSLスペクトルの高い周波数部分を占める。下流方向と上流方向の間の周波数分離は、高次バンドパスフィルタの使用に基づいている。図68は、上流周波数帯域及び下流周波数帯域スペクトルの間にガード帯域の使用を示す。さらに各下流スペクトルの帯域幅はモデムが異なると異なることができることを示す。これは、スペクトル割当てが、個々の回線状態及び下流から上流へのスループット比に従って最適化される故に必要なである。

【0241】バンドパスフィルタ・ストップバンドにおける有限の量の減衰及び下流スペクトルと上流スペクトルの接近の故に、反対チャネルからの多少の残留雑音が常にある。大きい加入者回線の減衰に起因して、相対的残留雑音の強さは、受信信号のそれに比較して無視できないことがある。異なるMDSL回線間での上流スペクトルと下流スペクトルの重なりに起因して、ガードバンドの領域内でNEXT雑音が発生し得る。従って、同じMDSL回線の反対チャネル残留雑音の干渉及び隣接するMDSL回線の反対チャネルNEXT雑音の干渉を最少にするため、NEXT打ち消しを使用することができる。

【0242】図69は、同じMDSLモデムブールユニット内で、追加のDSPチップをもち又はなしで、反対チャネルNEXTキャンセルバンクが実施できることを示す。NEXTキャンセルバンクは、送信信号及び全モデムのデジタル受信信号へのアクセスが必要である。NEXTキャンセルバンクは、図70に示す様にN個のNEXTキャンセルをもち、N個のMDSLモデムに対応する。各キャンセルは、サイズMのN個の適合フィルタをもつ。全N個の適合フィルタの出力は、対応するモ

デムに対するNEXTキャンセル信号を形成するため適切に組み合わせられる。各適合フィルタは、図71に示す様に、受信信号及びNEXTキャンセル信号及び対応する送信信号の間の誤差信号を相関ベクトルとして適合される。

【0243】次の用語／定義が、ここでは使用された。
MDSL-中間帯域デジタル加入者回線
MDSL-C-中央局位置で走行するMDSL
MDSL-R-住宅位置で走行するMDSL
POTS-普通の古い電話サービス。電話呼びを行いまた受けるだけ。

NDIS-ネットワークデバイスインタフェース仕様書。マイクロソフト（Microsoft）により定義された仕様書で、ネットワークドライバが互いにまたオペレーティングシステムと相互作用するための標準インタフェースを規定する。

NIC-ネットワークインタフェースカード

WAN-広域ネットワーク

ミニポートNICドライバ-NDIS3.0仕様書の拡充として開発されたネットワークインタフェースカードドライバで、デベロッパが、そのハードウェアに特定のコードだけを書き込み、共通事項をNDISライブラリ又はラツパに合併できるようにした。

【0244】以上の説明に関して更に以下の項を開示する。

(1) 受信したDSLモデム信号を復調する方法であつて、上記受信信号をバンドパスフィルタを通すステップと、上記受信信号を復調するため、上記受信信号を第1の高い周波数帯域からダウンサンプルして、第2の低い周波数帯域における上記信号の画像を作るステップと、からなる受信したDSLモデム信号を復調する方法。

(2) DSLモデム信号を変調する方法であつて、上記信号を用意するステップと、上記信号を第1の低い周波数帯域においてアップサンプルして、第2の高い周波数帯域における上記信号の画像を作るステップと、上記高い周波数帯域で上記信号を送信するため、上記第2の高い周波数帯域における上記信号の画像を選択するためバンドパスフィルタを通すステップと、からなるDSLモデム信号を変調する方法。

(3) 信号をモデムで処理する方法であつて、上記モデムにおいて、DMT回線コードを採用するステップと、上記DMT回線コードのための多重段階変換の少なくとも1つの計算段階においてデータのスケーリングを変えるステップとからなる信号をモデムで処理する方法。

(4) モデムであつて、少なくとも1つの段階において可変的にスケールされたデータを使用する多重段階変換を採用するデジタル信号処理装置からなるモデム。

【0245】(5) マルチモードモデムであつて、音声帯域モデムと、DSLモデムと、からなるマルチモード

モデム。

(6) 第5項記載のモデムであつて、さらにISDNインタフェースを含むマルチモードモデム。

(7) 第5項記載のモデムであつて、さらにスピーカフォンを含むマルチモードモデム。

(8) 第5項記載のモデムであつて、上記DSLモデムは、QAM、CAP、PSK、FSK、FM、AM、PAM、DTM、及びDWTMからなる群から選ばれた変調技術を使用するマルチモードモデム。

(9) 第5項記載のモデムであつて、上記DSLモデムは、対称的(下流帯域幅は上流帯域幅に等しい)であるマルチモードモデム。

(10) 第5項記載のモデムであつて、上記DSLモデムは、非対称(下流帯域幅は上流帯域幅と異なる)であるマルチモードモデム。

(11) マルチモードモデムであつて、音声帯域モデムと、DSLモデムとからなり、両方のモデムは、単一のマイクロプロセッサ又はマイクロコントローラ及びDSPを含むマルチモードモデム。

(12) 第11項記載のモデムであつて、上記DSPは、ASIC、ハイブリッドIC、又はVLSIチップの一部であるマルチモードモデム。

(13) 第11項記載のモデムであつて、さらにRISC又はマルチメディア命令セットを含むマルチモードモデム。

【0246】(14) マルチモードモデムであつて、音声帯域モデムと、DSLモデムとからなり、両方のモデムは、単一のDSP及びアナログ全面端を含むマルチモードモデム。

(15) 第14項記載のモデムであつて、上記アナログ全面端は、ID/A及びA/D回路を含むマルチモードモデム。

(16) 第14項記載のモデムであつて、上記アナログ全面端は、ID/A及びA/D及びフィルタ、隔離回路、AGC回路、及びコモンモード拒否回路を含むマルチモードモデム。

(17) マルチモードモデムであつて、音声帯域モデムと、DSLモデムとからなり、両方のモデムは、単一のDSP及びホストインタフェースを含むマルチモードモデム。

(18) マルチモードモデムであつて、音声帯域モデムと、DSLモデムとからなり、上記DSLモデムは、回線コードを選択する能力があるマルチモードモデム。

(19) モデムであつて、DSPと、上記DSPに接続されるA/D及びD/A回路と、上記A/D及びD/A回路に接続されるVSアナログ全面端と、上記A/D及びD/A回路に接続されるCAPアナログ全面端とからなるモデム。

(20) 第19項記載のモデムであつて、さらに、上記アナログ全面端に接続されるスプリッタを含むモデム。

(21) 第19項記載のモデムであつて、さらに、ROM、RAM、EPROMFlash、SRAM又はEEPROMの1つ又はそれより多くからなるメモリを含み、これは上記DSPに接続されるモデム。

(22) 第19項記載のモデムであつて、さらに、上記DSPに接続されるホストインタフェース回路を含むモデム。

(23) 第22項記載のモデムであつて、さらに、上記ホストインタフェース回路に接続されるホストバスを含むモデム。

(24) 第19項記載のモデムであつて、さらに、上記DSPに接続されるマイクロコントローラを含むモデム。

(25) 第19項記載のモデムであつて、さらに、上記DSPに接続されるLANインタフェース回路を含むモデム。

(26) 第19項記載のモデムであつて、さらに、上記DSPに接続されるWANインタフェース回路を含むモデム。

【0247】(27) マルチリンクモードモデムであつて、(i) 専用回線に関連する単一のリンクと、(ii) 専用回線に関連するマルチリンクと、(iii) ソフトダイヤルをもつ交換回線に関連する少なくとも単一のリンクと、(iv) ハードダイヤルをもつ交換回線に関連する少なくとも単一のリンクと、からなるマルチリンクモードモデム。

(28) モデムのためのリンク接続管理プロセスであつて、(i) リンク接続を確立するステップと、(ii) リンク切断を確立するステップと、(iii) リンク状態を監視するステップと、(iv) リンク状態をホスト処理装置に通知するステップと、(v) リンク構成、リンク接続/切断を上記2つのモデム間でメッセージで知らせるステップと、からなるモデムのためのリンク接続管理プロセス。

(29) 第28項記載のプロセスであつて、リンク接続確立の上記ステップ(i)は、更に、

a) 上記モデムに関連するホスト処理装置は、適当なパラメータを持つ回線接続コマンドを発行するサブステップと、

b) 上記モデムは、第2のモデムへリンク構成パケットを送るサブステップと、

c) 上記モデムは、上記第2のモデムからの応答を待つサブステップと、

d) もし、構成確認パケットを受信済みであれば、上記モデムは、リンク接続確立を通知するため割り込みを発生し、又はもし、構成が完全に拒否されていれば、回線構成は失敗し、又はもし、構成が部分的に拒否されていれば、もう1つの回線構成プロセスを開始するサブステップとからなるリンク接続管理プロセス。

【0248】(30) モデムを識別する方法であつて、上記モデムが接続されている回線のための電話番号を準

備するステップと、上記モデムを識別するため上記電話番号に1数字ポストフィックスを付加するステップと、からなるモデムを識別する方法。

(31) モデムのための速度交渉プロセスであつて、次のものに基づいてスループットを調節するステップを含み、(i) 上記モデムと第2のモデムの間の回線状態、

(i i) 主体ネットワークのアクセス可能性、I I I) 上記モデムの計算能力、及び(i v) アプリケーション要求事項を含むモデムのための速度交渉プロセス。

(32) 1対のDSLモデムにより使用される速度を選択する方法であつて、価格/性能モデムの範囲のなかで通信を可能にする相互に動作可能なモードを含む速度を選択する方法。

(33) モデムのための速度交渉プロセスであつて、回線において使用される帯域幅を調節し、また各記号により表されるビットの数を次のものに基づいて調節し、

(i) 上記モデムと第2のモデムの間の回線状態、(i i) 主体ネットワークのアクセス可能性、(i i i) 上記モデムの計算能力、及び(i v) アプリケーション要求事項、を含む速度交渉プロセス。

(34) 第33項記載のプロセスにおいて、上記モデムは、CAP回線コードを使用する速度交渉プロセス。

(35) 第33項記載のプロセスにおいて、上記モデムは、CAP回線コードを使用し、またアナログ全面端は帯域幅を調節できる速度交渉プロセス。

(36) 第33項記載のプロセスにおいて、上記モデムは、DMT回線コードを使用する速度交渉プロセス。

(37) 第33項記載のプロセスにおいて、上記モデムは、CAP及びDMT回線コードを使用する速度交渉プロセス。

(38) 第33項記載のプロセスにおいて、上記モデムは、QAM回線コードを使用する速度交渉プロセス。

【0249】(39) モデムチャネルマップメッセージであつて、(i) 全チャネル容量及び利用可能チャネル容量を示すチャネルマップヘッダ、(i i) 0又はそれより多いチャネルエントリにより表わされる現在のチャネル割当、(i i i) 1又はそれより多いチャネルエントリにより表わされるチャネル変更要求で、それらの最上位ビットによりフラグされる、モデムチャネルマップメッセージ。

(40) モデムのデータリンク層速度交渉の方法であつて、(i) 上記モデムはチャネルマップ変更要求メッセージを送り、そして応答を待つステップと、(i i) 関連するモデムは、チャネルマップ変更要求を調べ、そして次の可能性のあるメッセージを応答として送り出し、

- a) チャネルマップ変更要求を確認するためのチャネル変更要求確認メッセージ、
- b) チャネルマップ変更を拒否するためのチャネルマップ変更拒否、メッセージ、
- c) チャネルマップ変更再交渉の開始のためのチャネル

マップ変更ネガティブ確認メッセージ、を含むデータリンク層速度交渉の方法。

(41) 第40項記載の方法であつて、安定チャネル状態は次の1つにより達成され、

- a) チャネルマップ変更確認
- b) チャネルマップ変更拒否
- c) 応答タイムアウト待ち、又は
- d) チャネルマップ変更超過に対する最大の再試行、を含むデータリンク層速度交渉の方法。

【0250】(42) モデム使用CAP又はQAMの初期同期の方法であつて、既知の記号シーケンスの複数のセグメントを含む上記モデム及び第2のモデム間の立ち上げハンドシェイクのステップを含む、初期同期の方法。

(43) モデム使用CAP又はQAMの初期同期の方法であつて、サイクリック等化を使用して初期同期シーケンスを検出するステップと、そして初期同期シーケンスの反転バージョンからなる第2同期の検出によりタイミング瞬間を決定するステップと、からなるCAP又はQAMの初期同期の方法。

(44) モデムプールのNEXTキャンセラであつて、多重入力-多重出力(NINO)TEXTキャンセラバンクを含み、上記バンクは、多重入力NEXTキャンセラを含み、上記バンクは、上記モデムプールに接続され、また多重入力NEXTキャンセラは適合フィルタを含む、モデムプールNEXTキャンセラ。

(45) 第44項記載のキャンセラであつて、上記MIMO NEXTキャンセラは、上記モデムプール送信信号ベクトル及び上記モデムプール受信信号ベクトルを、NEXTキャンセル信号ベクトルを発生するための基準値として使用し、上記キャンセル信号ベクトルは正しく上記モデムプール受信信号ベクトルと組み合わせられ、上記モデムプールNEXT雑音レベルを減少させる、モデムプールNEXTキャンセラ。

(46) モデム初期化の方法であつて(i) 第1のそれぞれ間隔を置いたトーンで、その周波数は、チャネルを特徴づけるため音声帯域より高く、そして(i i) 第2のそれぞれ間隔を置いたトーンで、同じ周波数範囲にあつて、モデムの能力を決定し、そしてユーザの好みの情報を交換する、モデム初期化の方法。

【0251】(47) モデムのための速度交渉プロセスであつて、物理的送信スループット維持手順、フレーム及びタイムスロット割当手順ソフトウェア及びハードウェアインタフェース、及び状態遷移アルゴリズム、を含む速度交渉プロセス。

(48) 単一のDSPデバイス上で同時に動作する多重モデムからなるシステム。

(49) 第48項記載のシステムであつて、上記モデムの複数の1つは、音声帯域より上で動作する、システム。

(50) DMTモデム初期同期の方法であつて、(i) ブロックの極性が周期的に交番する反復的データブロックを含むチャンネルインプルス応答の長さを調べるための訓練シーケンス、(ii) 前に受信したデータブロックを現在受信したデータブロックから差し引くことにより、残留チャンネル記号間干渉及びフレーム境界を求め、(iii) 等化器の数を更新するため、上記訓練シーケンス及び上記残留チャンネル記号間干渉を使用することにより、時間領域等化器を訓練し、そして(iv) 上記訓練シーケンス及び上記フレーム境界を使用してフレーム同期をする、初期同期の方法。

(51) 上記訓練シーケンスを検出する方法であつて、現在受信しているデータブロックのパワーを、現在受信しているデータブロックを前に受信したブロックに加えることにより構成されるデータブロックパワート比較する、訓練シーケンスを検出する方法。

(52) テレビ会議システムであつて、ビデオは、音声帯域より高い周波数を使用するモデムにより送信され、また音声は音声帯域で同じ加入者回線上を送信されるシステムからなる、テレビ会議システム。

【0252】(53) モデムハードウェアをホストオペレーティングシステムとインタフェースする方法であつて、

- a) 定義された1組のホストインタフェース機能を、コマンド制御のため呼び出すステップと、
- b) 定義された1組のホストインタフェース機能を、リンク接続管理のため呼び出すステップと、
- c) 定義された1組のホストインタフェース機能を、データ送信/受信のため呼び出すステップと、
- d) 定義された1組のホストインタフェース機能を、音声帯域の音声と音声帯域より高いビデオの同期のため呼び出すステップと、
- e) 定義された1組のホストインタフェース機能を、音声帯域を制御チャンネルとして使用し、音声帯域の上の部分をデータチャンネルとして使用するため呼び出すステップと、からなるインタフェースする方法。

(54) マルチモードモデムのためのモデムの新しいモードを確立する方法であつて、追加のモードを支持するためにモデム能力を向上させるため、利用できるモデムモードをもつソフトウェアを受けとるステップを含む、新しいモードを確立する方法。

【0253】(55) 音声帯域の周波数帯域とより高い周波数帯域とで選択的に動作するモデムが提供される。このモデムは、DMT及びCAPの様な多重回線コードを、支援する。このモデムは、離散マルチトーン(DMT)及び搬送波無しAM/PM(CAP)の様な異なる現存するADSL回線コードが、同じハードウェアプラットフォーム上で実施できる様に、デジタル信号処理装置(DSP)を使用する。モデムは、回線状態及びサービス費用要求を調停するため、希望する伝送速度

に対して実時間で交渉する。回線コード及び速度交渉プロセスは、各通信の初めにモデム間で、トーンの交換を介して実施される。4ステップMDSL初期化プロセスが、回線コードと速度の両立性のために規定されている。CAP使用モデムもための、新しい同期立ち上げ手順が、定められている。ハンドシェイクプロトコル及び受信機アルゴリズムは、標準撚線対電線の様な、酷く振幅の歪んだチャンネルを通じて、信頼性のあるモデム同期を可能にする。またアルゴリズムは、受信機において、同期等化器を訓練するため、短い長さのシーケンスを使用する。このシーケンスの訓練の後、整合されたフィルタ又は相関器が反転された同期シーケンスを検出するため使用される。この反転シーケンスの検出が、CAP復調等化器の通常の基準訓練の開始の合図となる。MDSL回線接続管理プロセスは、電気通信広域ネットワーク環境において、MDSL-C(中央局のMDSL)とMDSL-R(住宅位置のMDSL)の間の回線接続を管理するための簡単で、効率的な、そして柔軟性あるインタフェースを提供する。MDSLモデム内の内部状態機械は回線状態を記録し、監視し、そして状態変化を他のモデム及びホスト処理装置にも通知する。回線接続管理メッセージの交換に使用されるプロトコルは、MDSLのために簡略化されたリンク制御プロトコル(LCP)である。

【0254】

【表8】

APPENDIX

件録

[0255]

[表9]

```

@INPUT_DECLARATIONS:
  STATIC Double I_clock;
  STATIC Ovector I_real_in;
  STATIC Double I_reset;
  STATIC Double I_valid_in;
@OUTPUT_DECLARATIONS:
  STATIC Ovector O_img_out;
  STATIC Ovector O_real_out;
  STATIC Double O_scale;
  STATIC Double O_valid_out;
@PARAMETER_DECLARATIONS:
{
  STATIC Long P_fft_math;
}
@STATE_DECLARATIONS:
{
  STATIC Long size ;
  STATIC Double *f_real ;
  STATIC Double *f_img ;
  STATIC short *i_real ;
  STATIC short *i_img ;
  STATIC long *l_real ;
  STATIC long *l_img ;
  STATIC Double *in_r ;
  STATIC Double *out_i ;
  STATIC Double *out_r ;
  STATIC long flag ;
  STATIC long clock ;
  STATIC long ltemp ;
  STATIC long lmax ;
  STATIC int scale ;
  STATIC int i ;
  STATIC int j ;
  STATIC int l ;
  STATIC int max_bttr ;
  STATIC int j2 ;
  STATIC int j11 ;
  STATIC short temp ;
  STATIC int k ;
  STATIC double twopi ;
  STATIC double tmp_real ;
  STATIC double tmp_img ;
  STATIC int doit ;
  STATIC int Sine[256] ;
  STATIC int Cosine[256] ;
  STATIC FILE * file1 ;
  STATIC int lmm ;
  STATIC int lix ;
  STATIC double scl;
  STATIC int bttr;
  STATIC int li ;
  STATIC int power ;
  STATIC int stage ;
  STATIC double t1 ;
  STATIC double t2 ;
  STATIC double arg ;
  STATIC double c ;
  STATIC double s ;
  STATIC int it1 ;
  STATIC int it2 ;
  STATIC int ic ;
  STATIC int is ;

```

```

STATIC int  arginc ;
STATIC int  iarg ;
STATIC short tmp_r ;
STATIC short tmp_i ;
STATIC int  itmpire ;
STATIC int  itmplim ;
STATIC int  itmp2re ;
STATIC int  itmp2im ;
STATIC double pi ;
STATIC double c1 ;
STATIC double c2 ;
STATIC double wpr ;
STATIC double wpi ;
STATIC double wi ;
STATIC double wr ;
STATIC double wtmp ;
STATIC double tmpire ;
STATIC double tmpim ;
STATIC double tmp2re ;
STATIC double tmp2im ;
STATIC int  length ;
STATIC int  half ;
STATIC int  halfpow ;
STATIC int  DEBUG ;
STATIC long  lt1 ;
STATIC long  lt2 ;
STATIC long  ls ;
STATIC long  lc ;
STATIC long  ltemp1 ;
STATIC long  ltemp2 ;
STATIC long  ltemp3 ;
STATIC long  ltemp4 ;
STATIC long  l_tmp_r ;
STATIC long  l_tmp_i ;
STATIC long  ltmpire ;
STATIC long  ltmplim ;
STATIC long  ltmp2re ;
STATIC long  ltmp2im ;
STATIC double ftemp ;
}
@BLOCK_DECLARATIONS:
{
}
@INITIALIZATION_CODE:
{
DEBUG = 0 ;
size = 512 ;
power = 9 ;
if(P_fft_math == 1 )
{
f_real = (double *) malloc (size * sizeof(double)) ;
f_img = (double *) malloc (size * sizeof(double)) ;
}
else
{
i_real = (short *) malloc (size * sizeof(short)) ;
i_img = (short *) malloc (size * sizeof(short)) ;
l_real = (long *) malloc (size * sizeof(long)) ;
l_img = (long *) malloc (size * sizeof(long)) ;
}
twopi = 3.1415926536 * 2.0 ;
pi = 3.1415926536 ;
file1 = fopen("/home/mannerin/sincos1.txt", "r") ;
for(j=0 ; j < 256 ; j++)

```

[0256]

[表10]


```

    fscanf(file1, "%08x \n", &Cosine[j]);
}
for(j=0; j < 256; j++)
{
    fscanf(file1, "%08x \n", &Sine[j]);
}
fclose(file1);
}
@RUN_OUT_CODES: (実行コードの開始)
/* start of run code */
    flag = 0;
    O_valid_out = 1.0;
    O_scaler = 0.0;
    @IF (I_clock == CONNECTED)
        clock = (long) I_clock;
        flag |= (~clock & 1);
    @ENDIF
    @IF (I_valid_in == CONNECTED)
        flag |= (long) I_valid_in;
    @ENDIF
    @IF (I_reset == CONNECTED)
        if(I_reset == 1.0)
        {
            flag = 0;
        }
    @ENDIF
    if(!flag)
    {
        /* start of process input/output */
        /* read in input */
        in_r = (double *) OvGetStart(I_real_in);
        out_i = (double *) OvGetStart(O_img_out);
        out_r = (double *) OvGetStart(O_real_out);
        size = 512;
        power = 9;
        if(P_fft_math == 1)
        {
            /* start of if floating */ if floatingの開始
            for(i=0; i < size; i++)
            {
                f_real[i] = in_r[i];
            }
        }
        /* start of do floating fft */ do floating fftの開始
        length = 512;
        half = 256;
        halfpow = power - 1;
        c1 = c2 = 0.5;
        for(i = 0, j = 0; i < half; i++, j += 2)
        {
            f_real[i] = in_r[j];
            f_img[i] = in_r[j + 1];
        }
        c2 = -c2;
        /* start of do floating fft */
        size = 256;
        power = 8;
        max_bttr = size;
        for(stage = 1; stage <= power; stage++) {
            lxx = max_bttr;
            max_bttr /= 2;
            lmm = max_bttr;
            scl = ((double) (-twopi) / (double) lxx);
            for(bttr = 1; bttr <= lmm; bttr++) {
                arg = (bttr - 1) * scl;
            }
        }
    }
}

```

[0257]

【表11】

```

    c = cos (arg) ;
    s = sin (arg);
    for(li = lix; li <= size ; li += lix) {
        j11 = li - lix + btttr - 1;
        j2 = j11 + max btttr;
        t1 = f_real[j11] - f_real[j2] ;
        t2 = f_img[j11] - f_img[j2] ;
        f_real[j11] = f_real[j11] + f_real[j2] ;
        f_img[j11] = f_img[j11] + f_img[j2] ;
        f_real[j2] = (c*t1 + s*t2);
        f_img[j2] = (c*t2 - s*t1);
    }
}

j = 0 ;
for(i = 0; i < (size - 1); i++) {
    if(i < j) {
        tmp_real = f_real[j] ;
        tmp_img = f_img[j] ;
        f_real[j] = f_real[i] ;
        f_img[j] = f_img[i] ;
        f_real[i] = tmp_real;
        f_img[i] = tmp_img;
    }
    k = (size / 2);
    while (k < (j + 1)) { j -= k; k /= 2; } ... (偶数/奇数分離アルゴリズム)
    j += k;
}

/* Perform even/odd separation algorithm.
/* Form an array with N/2 + 1 elements.
*/
arg = pi / half;
wtmp = sin (0.5 * arg);
wpr = - 2.0 * wtmp * wtmp;
wpi = sin (arg);
wr = 1.0 + wpr;
wi = wpi;
for (i = 1, j = (half - 1); i <= j; i++, j--)
{
    tmp1re = (f_real[i] + f_real[j]) * c1;
    tmp1im = (f_img[i] - f_img[j]) * c1;
    tmp2re = (f_img[i] + f_img[j]) * -c2;
    tmp2im = (f_real[i] - f_real[j]) * c2;
    f_real[i] = (tmp1re + wr * tmp2re - wi * tmp2im);
    f_img[i] = (tmp1im + wr * tmp2im + wi * tmp2re);
    f_real[j] = (tmp1re - wr * tmp2re + wi * tmp2im);
    f_img[j] = (- tmp1im + wr * tmp2im + wi * tmp2re);
    wtmp = wr;
    wr += (wr * wpr - wi * wpi);
    wi += (wi * wpr + wtmp * wpi);
}

/* Compute the first pair of frequency cells,
/* at the dc and the Nyquist point
*/
wr = f_real[0] ;
f_real[0] = wr + f_img[0] ;
f_img[0] = f_img[0] - wr ;

for(i=0 ; i < size ; i++)
{
    out_r[i] = f_real[i] ;
}

```

(偶数/奇数分離アルゴリズム)の
 A 実行 (N/2+1要素を
 配列A形成)

同法でセルの第1ペアと計算

dc 及び ナイキスト点において

```

        out_i[i] = -f_img[i];
    }
} /* end of if floating */
/*****
else
    if (P_fft_math == 2)
    {
        /* start of int fft */
        length = 512;
        half = 256;
        halfpow = power - 1;
        for (i = 0, j = 0; i < half; i++, j += 2)
        {
            l_real[i] = (long) (in_r[j] * 65536.0);
            l_img[i] = (long) (in_r[j + 1] * 65536.0);
        }
        lmax = 0;
        for (i = 0; i < half; i++)
        {
            ltemp = l_real[i];
            if (ltemp < 0) ltemp = -ltemp;
            if (ltemp > lmax) lmax = ltemp;
            ltemp = l_img[i];
            if (ltemp < 0) ltemp = -ltemp;
            if (ltemp > lmax) lmax = ltemp;
        }
        scale = 0;
        for (i = 0; i < 32; i++)
        {
            lmax <= 1;
            if (lmax < 0)
                break;
            else
                scale++;
        }
        scale--;
        scl = 1.0;
        for (i = 0; i < scale; i++)
            scl *= 2.0;
        lmax = 65536 / (long) scl;
        for (i = 0; i < half; i++)
        {
            i_real[i] = (short) (l_real[i] / lmax);
            i_img[i] = (short) (l_img[i] / lmax);
        }
        /* to fft */
        size = 256;
        power = 8;
        arginc = 2;
        max_btr = size;
        scale = 0;
        for (stage = 1; stage <= power; stage++) {
            doit = 0;
            for (j = 0; j < size; j++)
            {
                temp = (i_real[j] & (short) 0xc000);
                if ((temp != 0) && (temp != (short) 0xc000))
                {
                    doit = 1; break;
                }
                temp = (i_img[j] & (short) 0xc000);
                if ((temp != 0) && (temp != -(short) 0xc000))
                {
                    doit = 1; break;
                }
            }
        }
        if (doit)
        {
            scale++;
        }
    }
}

```

if floating 32
int fft 開始
fft へ移行

```

for(j=0 ; j < size ; j++)
{
    i_real[j] /= 2 ;
    i_img[j] /= 2 ;
}
}
for(j=0 ; j < size ; j++)
{
    temp = (i_real[j] & (short) 0xe000) ;
    if((temp != 0) && (temp != (short) 0xe000))
    {
        doit = 1 ; break ;
    }
    temp = (i_img[j] & (short) 0xe000) ;
    if((temp != 0) && (temp != (short) 0xe000))
    {
        doit = 1 ; break ;
    }
}
if(doit)
{
    scale++ ;
    for(j=0 ; j < size ; j++)
    {
        i_real[j] /= 2 ;
        i_img[j] /= 2 ;
    }
}
iarg = 0 ;
lix = max_bttr ;
max_bttr /= 2 ;
lmm = max_bttr ;
for(bttr = 1 ; bttr <= lmm ; bttr++) {
    ic = (int) Cosine[iarg] ;
    is = (int) - Sine[iarg] ;
    iarg += arginc ;
    for(li = lix ; li <= size ; li += lix) {
        j11 = li - lix + bttr - 1 ;
        j2 = j11 + max_bttr ;
        it1 = i_real[j11] - i_real[j2] ;
        it2 = i_img[j11] - i_img[j2] ;
        i_real[j11] = i_real[j11] + i_real[j2] ;
        i_img[j11] = i_img[j11] + i_img[j2] ;
        i_real[j2] = (short) (((ic*it1) + (is*it2)) / 32768) ;
        i_img[j2] = (short) (((ic*it2) - (is*it1)) / 32768) ;
    }
}
arginc *= 2 ;
}
j = 0 ;
for(i = 0 ; i < (size - 1) ; i++) {
    if(i < j) {
        tmp_r = i_real[i] ;
        tmp_i = i_img[i] ;
        i_real[j] = i_real[i] ;
        i_img[j] = i_img[i] ;
        i_real[i] = tmp_r ;
        i_img[i] = tmp_i ;
    }
    k = (size / 2) ;
    while (k < (j + 1)) { j -= k ; k /= 2 ; }
    j += k ;
}
/*****
/* Perform even/odd separation algorithm.
/* Form an array with N/2 elements.
*****/
k = 1 ;
for (i = 1, j = (half - 1) ; i <= j ; i++, j--)
{

```

偶数/奇数分離
7/2, 1/2, 3/2, 5/2の進行
N/2+1の要素の並びの
形成

```

/*      wr => Costable[1..] wi => Sintable[1..] */
is = Sine[k] ;
ic = Cosine[k] ;
itmplre = (int)((i_real[i] + i_real[j]) >> 1) ;
itmplim = (int)((i_img[i] - i_img[j]) >> 1) ;
itmp2re = (int)((i_img[i] + i_img[j]) >> 1) ;
itmp2im = (int)((i_real[i] - i_real[j]) >> 1) ;
i_real[i] = (short) (itmplre + ((ic * itmp2re - is * itmp2im)
/ 32768));
i_img[i] = (short) (itmplim + ((ic * itmp2im + is * itmp2re)
/ 32768));
i_real[j] = (short) (itmplre - ((ic * itmp2re - is *
itmp2im) / 32768));
i_img[j] = (short) (- itmplim + ((ic * itmp2im + is *
itmp2re) / 32768));
k++ ;
}

/* *****
/* Compute the first pair of frequency cells, */
/* at the dc and the Nyquist point */
/* *****
is = i_real[0] ;
i_real[0] = is + i_img[0] ;
i_img[0] = i_img[0] - is ;

/* *****
scl = 1.0 ;
for(i=0 ; i < scale ; i++)
    scl *= 2.0 ;
for(i=0 ; i < size ; i++)
{
/*      out_r[i] = (double) (i_real[i] << scale) ;
/*      out_i[i] = (double) (-i_img[i] << scale) ;

/*
    out_r[i] = (double) (i_real[i]) ;
    out_i[i] = (double) (-i_img[i]) ;
    out_r[i] *= ((scl * (float)lmax)/65536.0) ;
    out_i[i] *= ((scl * (float)lmax)/65536.0) ;

/*
    out_r[i] *= scl/(float)lmax ;
    out_i[i] *= scl/(float)lmax ; /* int fft a 終了

} /* end of int fft */
O_valid_out = 0.0 ;
O_scaler = (double) scale ;
} /* end of flag */
} /* end of run code */
TERMINATION_CODE:
{
    if(P_fft_math == 1)
    {
        free(f_real) ;
        free(f_img) ;
    }
    else
    {
        free(i_real) ;
        free(i_img) ;
        free(l_real) ;
        free(l_img) ;
    }
}
}

```

同波数セルの対称計算
スカラー倍を省略して

int fft a 終了

flag a 終了

終了コードの終了

ク制御プロトコル (LCP) である。

【図面の簡単な説明】

- 【図1】 マルチモードモデムの好ましい実施例を示す。
- 【図2】 図1と同じ説明。
- 【図3】 図1と同じ説明。
- 【図4】 図1と同じ説明。
- 【図5】 図1と同じ説明。
- 【図6】 モデム中央局モデムの好ましい実施例を示す。
- 【図7】 図6と同じ説明。
- 【図8】 図6と同じ説明。
- 【図9】 モデムアプリケーション及びISDNシグナリ

ングの好ましい実施例を示す。

- 【図10】 図9と同じ説明。
- 【図11】 図9と同じ説明。
- 【図12】 図9と同じ説明。
- 【図13】 図9と同じ説明。
- 【図14】 従来技術のモデム及び加入者回線容量を示す。
- 【図15】 図14と同じ説明。
- 【図16】 図14と同じ説明。
- 【図17】 モデムの他の好ましい実施例を示す。
- 【図18】 図17と同じ説明。

【図19】初期化の好ましい実施例を示す。

【図20】図19と同じ説明。

【図21】図19と同じ説明。

【図22】図19と同じ説明。

【図23】図19と同じ説明。

【図24】図19と同じ説明。

【図25】速度交渉の好ましい実施例を示す。

【図26】図25と同じ説明。

【図27】図25と同じ説明。

【図28】図25と同じ説明。

【図29】図25と同じ説明。

【図30】図25と同じ説明。

【図31】同期化の好ましい実施例を示す。

【図32】図31と同じ説明。

【図33】図31と同じ説明。

【図34】訓練の好ましい実施例を示す。

【図35】図34と同じ説明。

【図36】図34と同じ説明。

【図37】図34と同じ説明。

【図38】回線接続管理の好ましい実施例を示す。

【図39】図38と同じ説明。

【図40】図38と同じ説明。

【図41】図38と同じ説明。

【図42】図38と同じ説明。

【図43】図38と同じ説明。

【図44】図38と同じ説明。

【図45】図38と同じ説明。

【図46】モデムドライバの好ましい実施例を示す。

【図47】図46と同じ説明。

【図48】図46と同じ説明。

【図49】図46と同じ説明。

【図50】図46と同じ説明。

【図51】図46と同じ説明。

【図52】図46と同じ説明。

【図53】図46と同じ説明。

【図54】図46と同じ説明。

【図55】図46と同じ説明。

【図56】図46と同じ説明。

【図57】図46と同じ説明。

【図58】図46と同じ説明。

【図59】ダウンロードの好ましい実施例を示す。

【図60】サンプリング速度変換の好ましい実施例を示す。

【図61】図60と同じ説明。

【図62】図60と同じ説明。

【図63】図60と同じ説明。

【図64】図60と同じ説明。

【図65】図60と同じ説明。

【図66】図60と同じ説明。

【図67】モデムプールの好ましい実施例を示す。

【図68】図67と同じ説明。

【図69】図67と同じ説明。

【図70】図67と同じ説明。

【図71】図67と同じ説明。

【符号の説明】

100 マルチモードモデム

110 VB AFE (音声帯域アナログフロントエンド)

120 DSL AFE (加入者回線アナログフロントエンド)

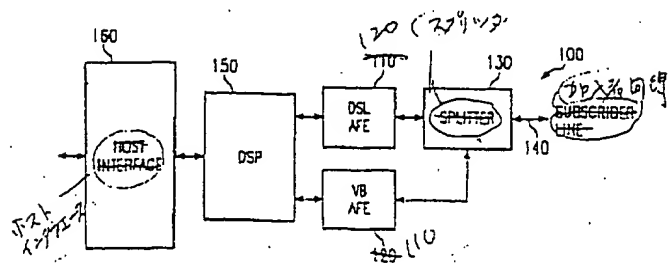
130 スプリッタ

140 加入者回線

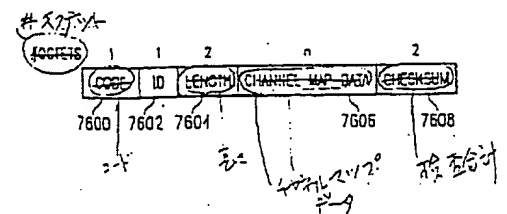
150 DSP (デジタル信号処理装置)

160 ホストインタフェース

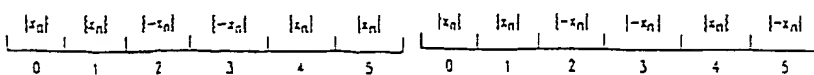
【図1】



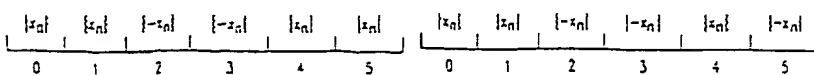
【図26】



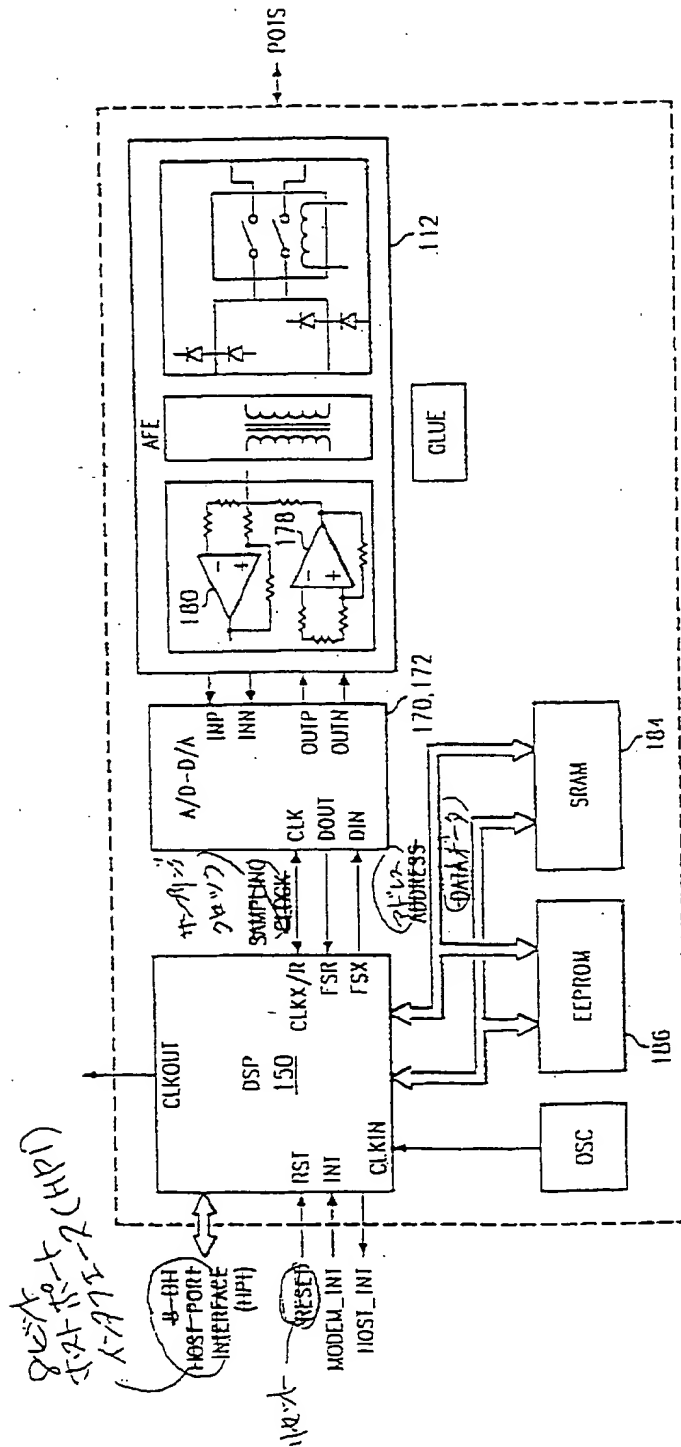
【図34】



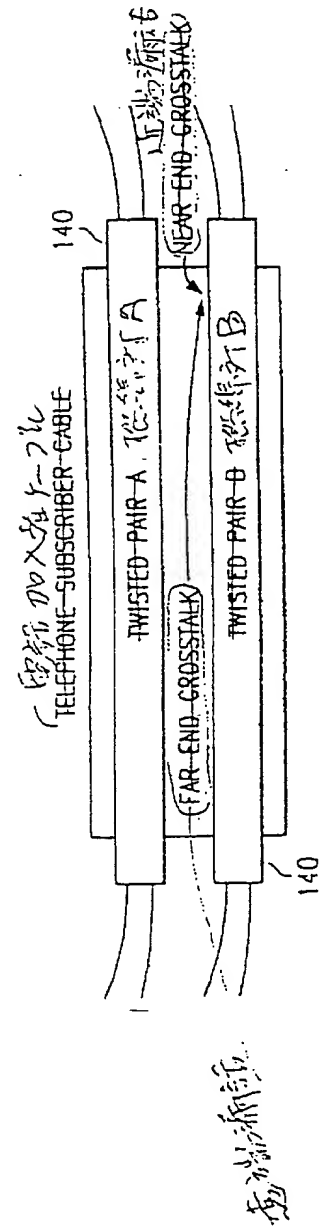
【図36】



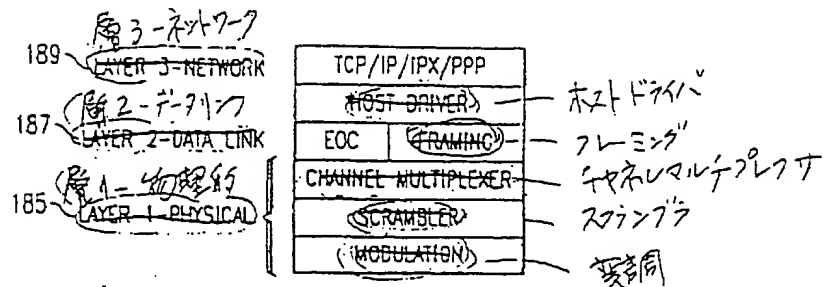
【図3】



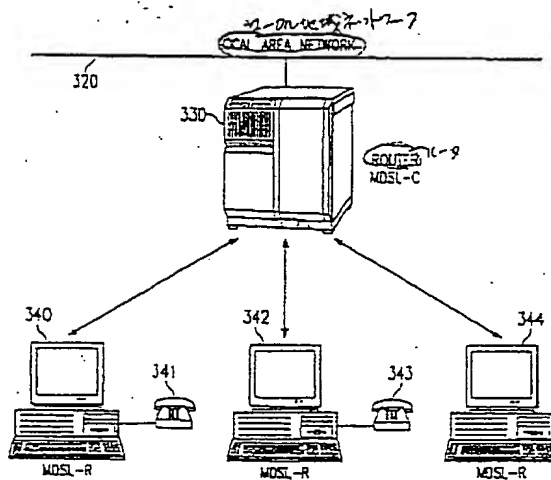
【図20】



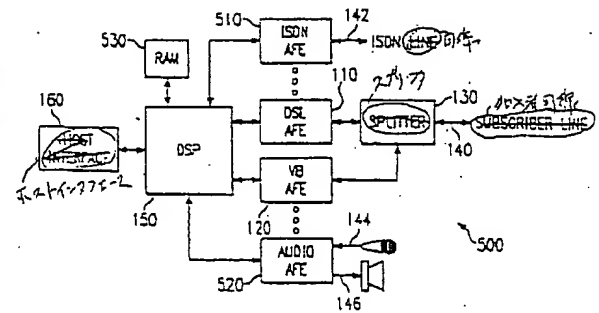
【図 5】



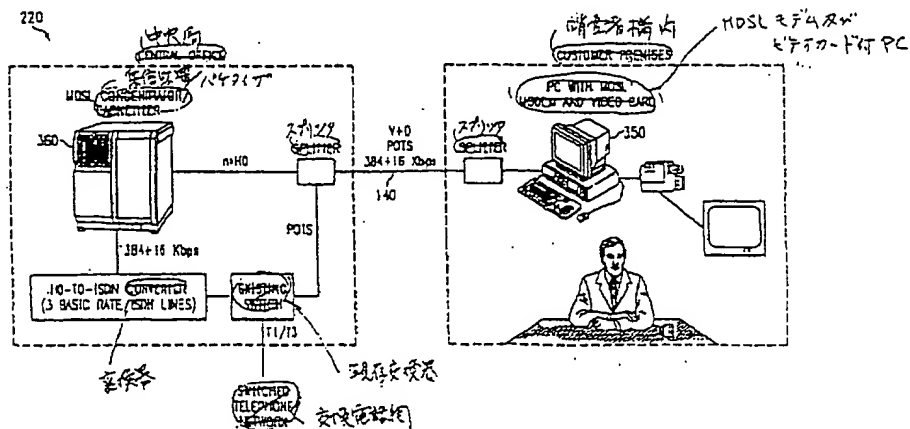
【図 10】



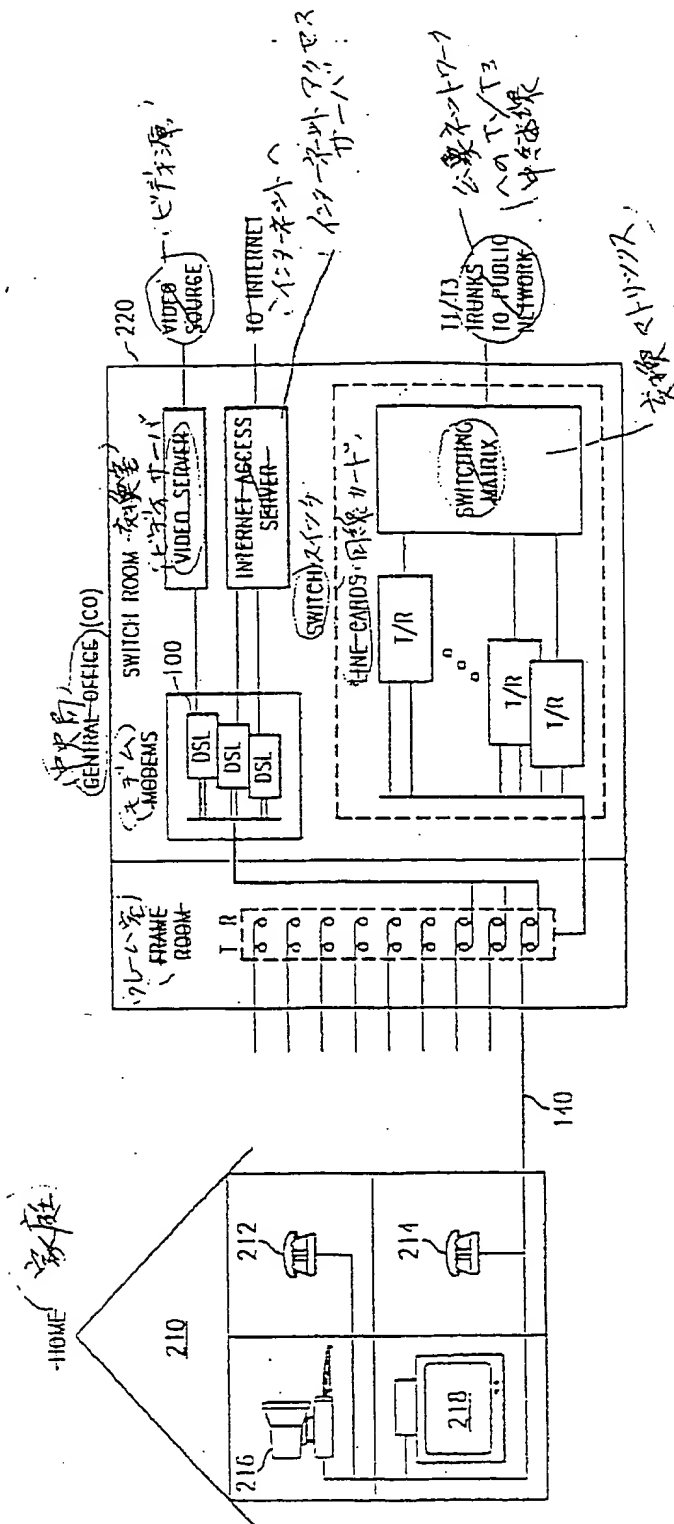
【图 17】



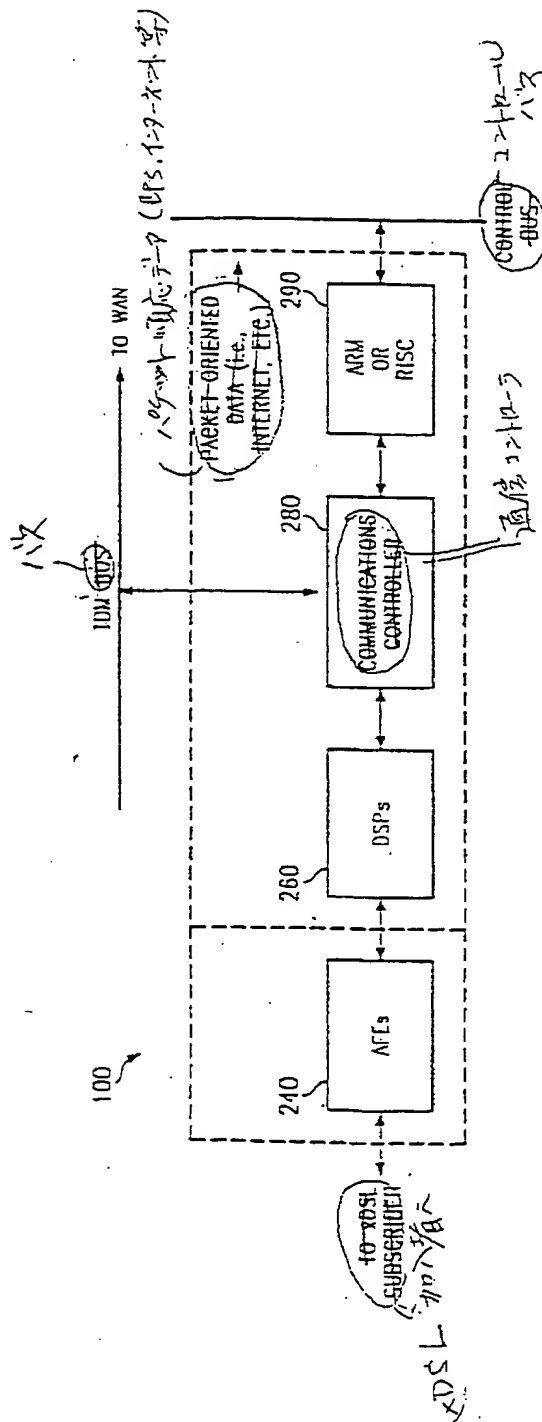
【図 11】



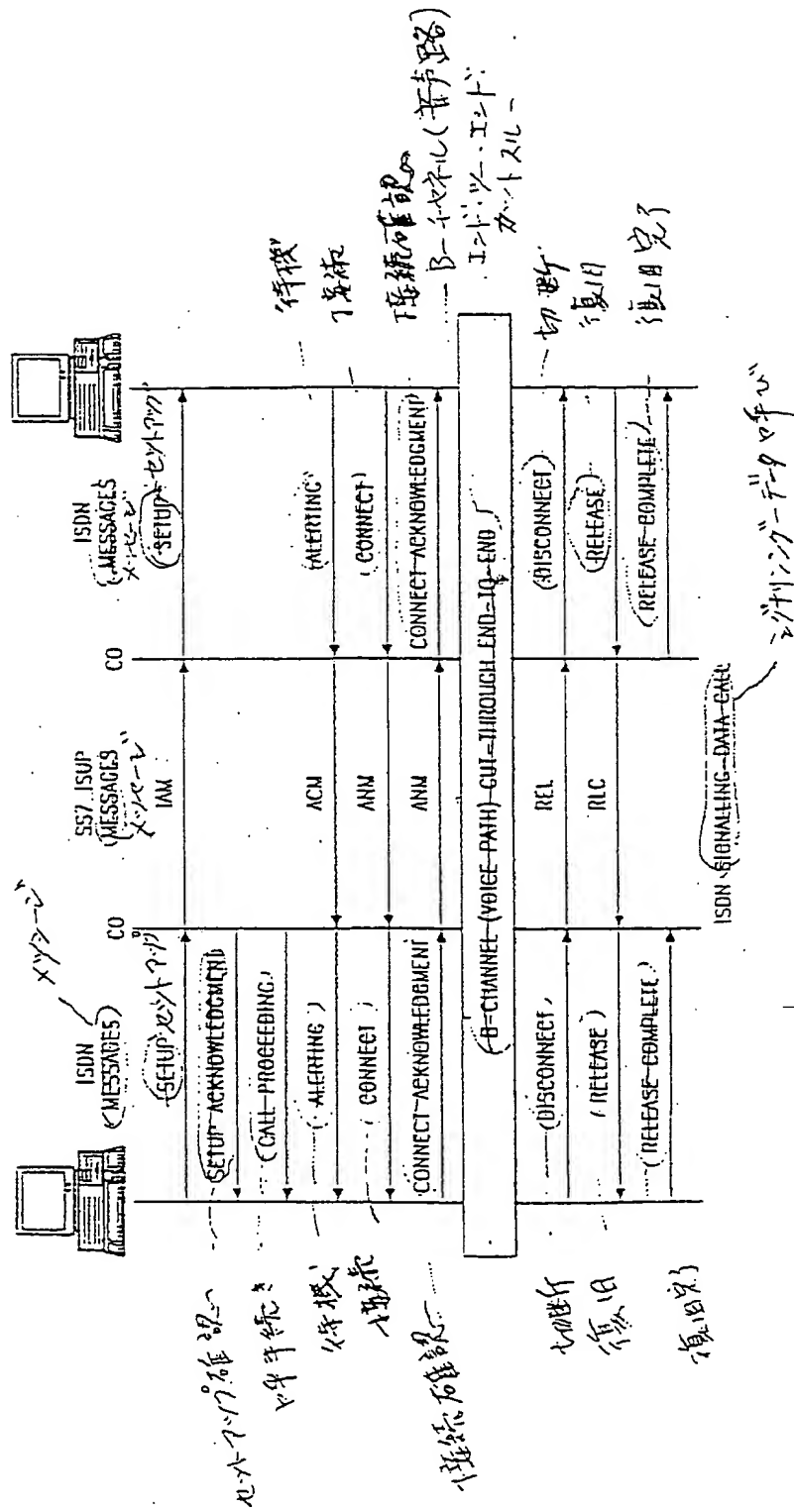
【図6】



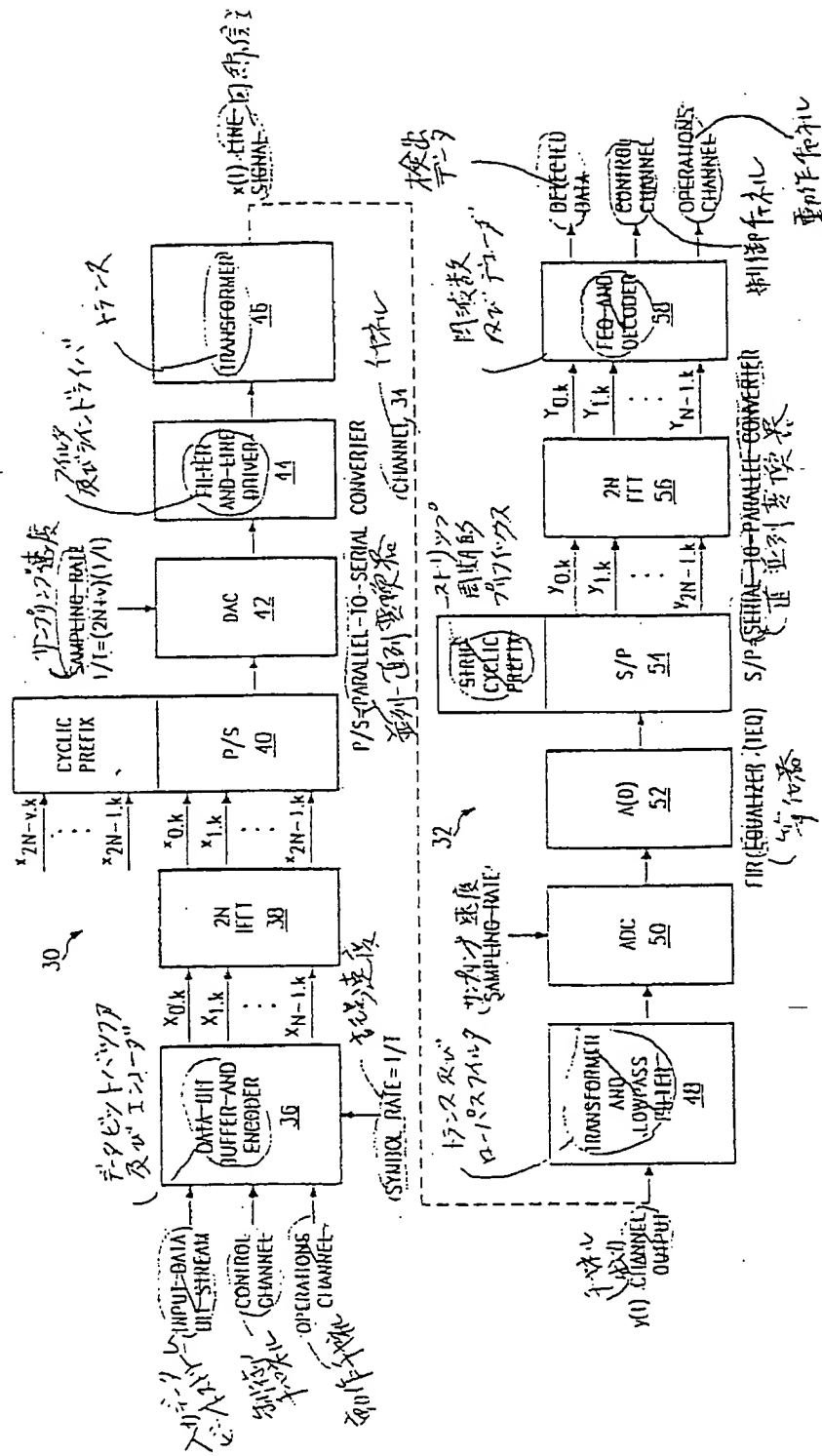
【図8】



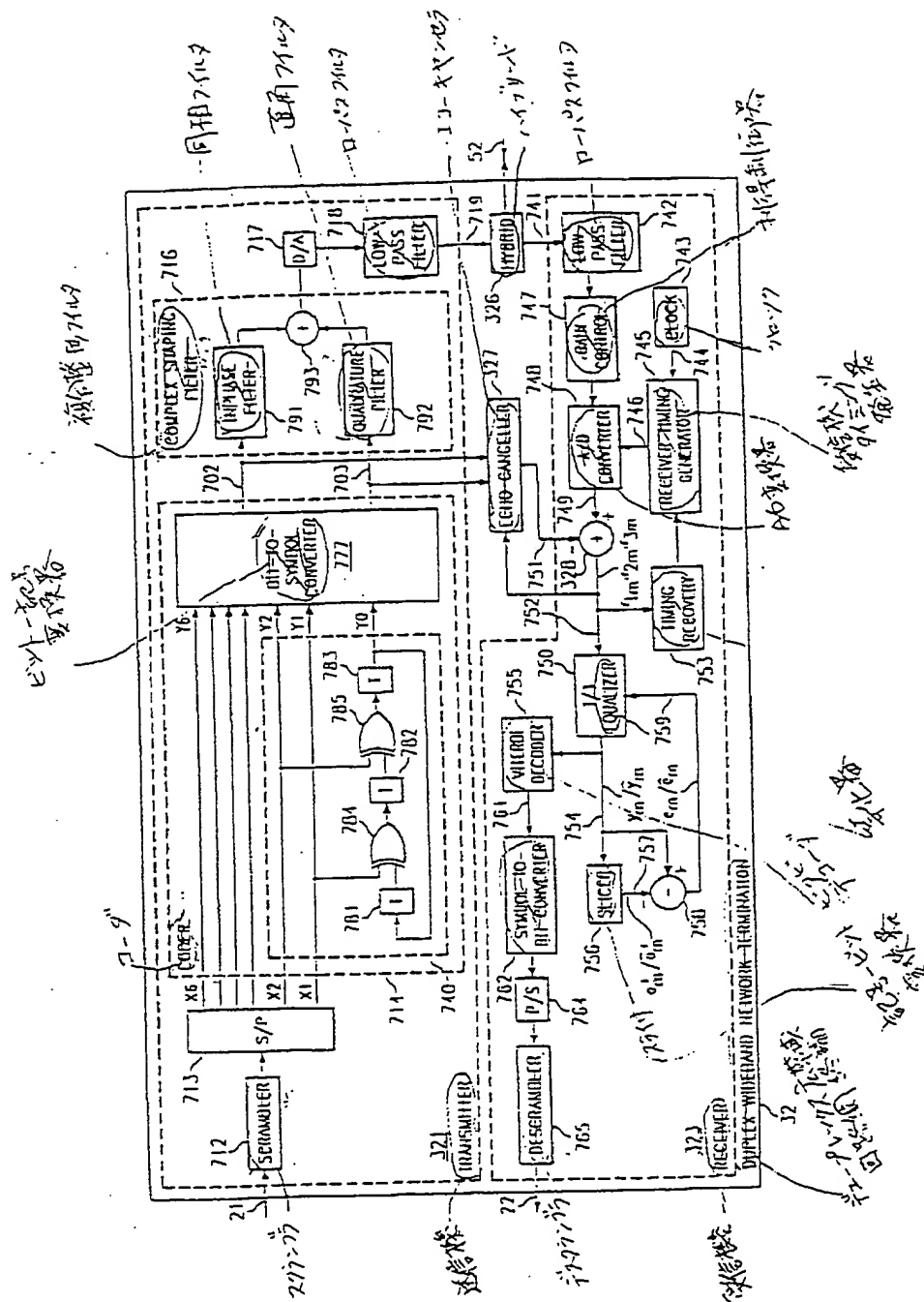
【図12】



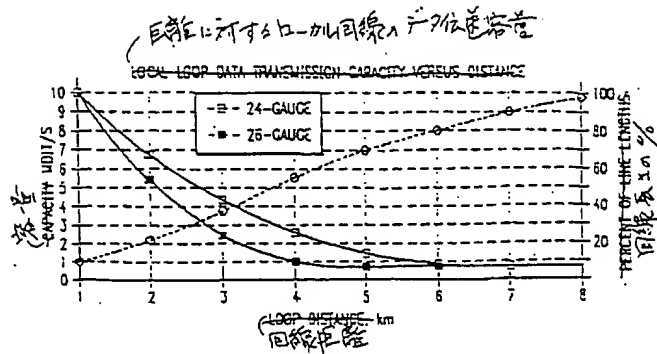
[図14]



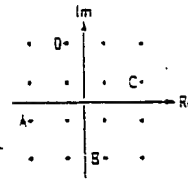
【图 15】



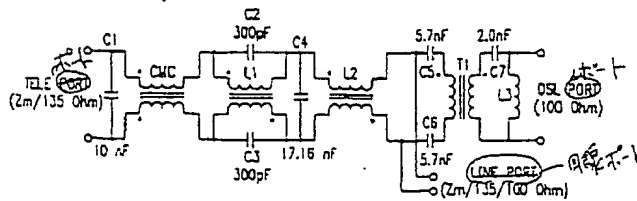
【図 16】



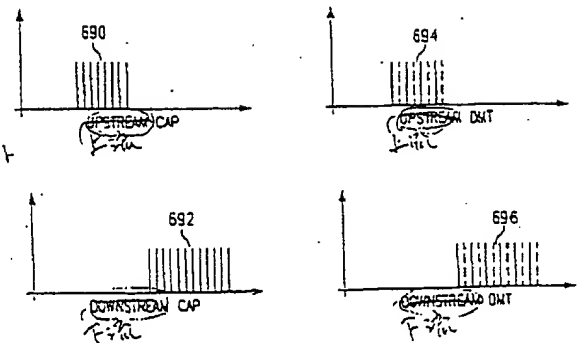
【図 3 1】



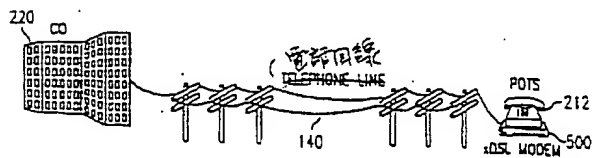
【图 18】



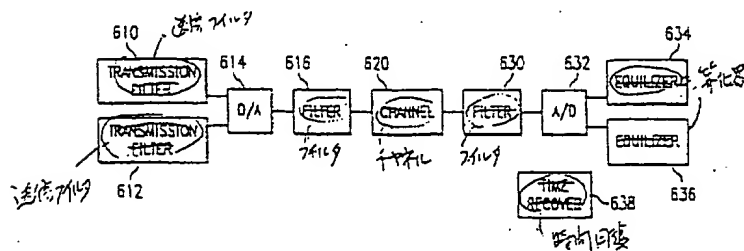
【图 24】



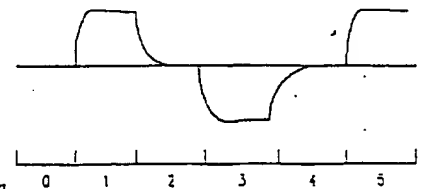
【例 19】



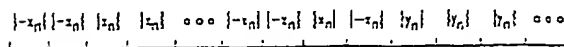
【図 2 1】



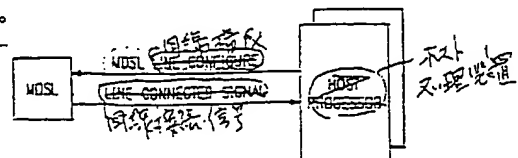
【图 35】



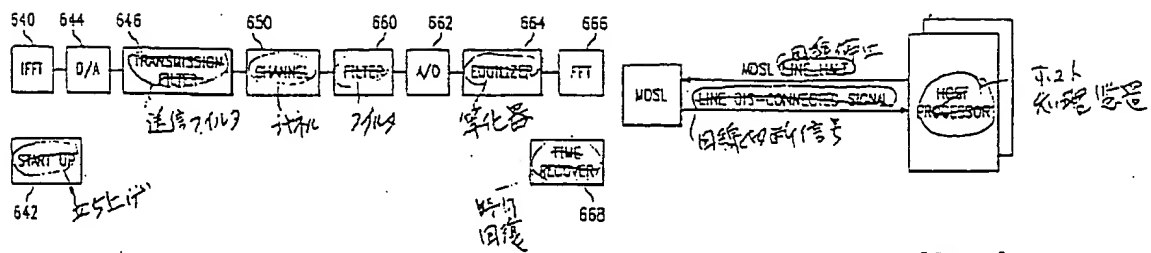
【图 3 7】



【图 38】

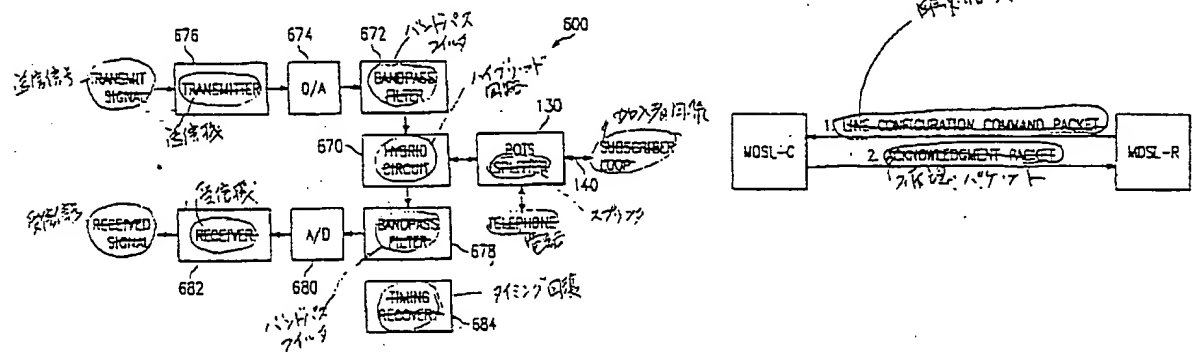


【☒ 39】

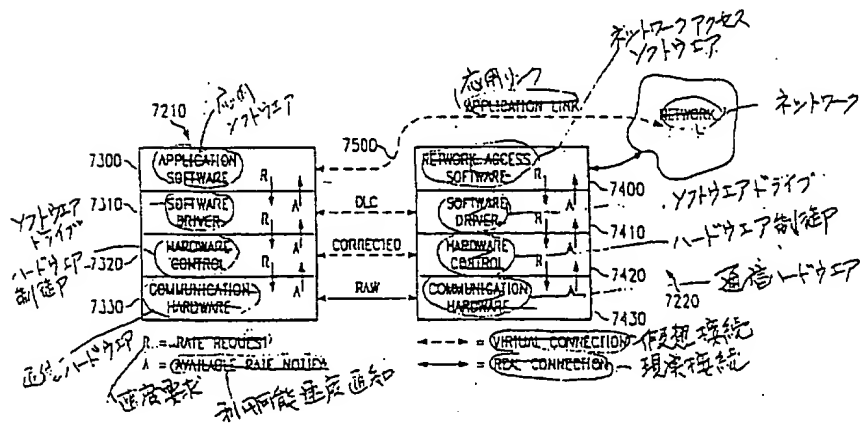


【图 4-4】

【图 2 3】

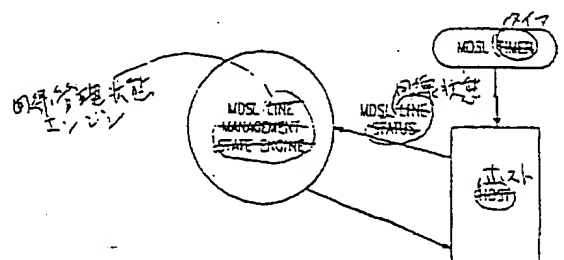
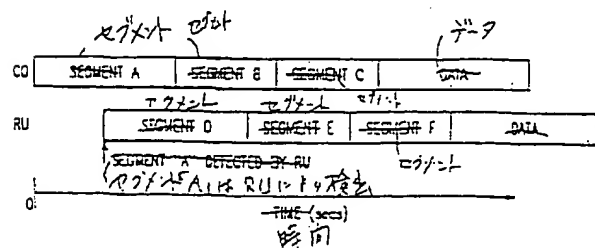


【图 25】

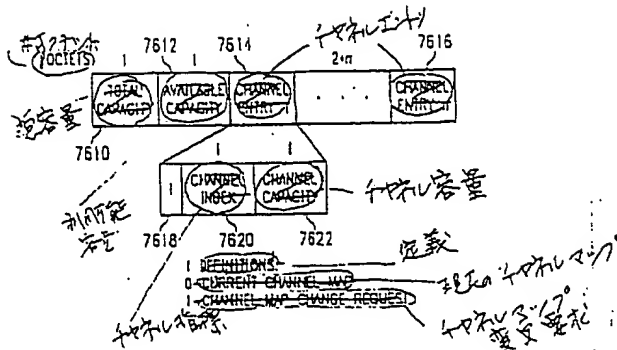


【图 3 3】

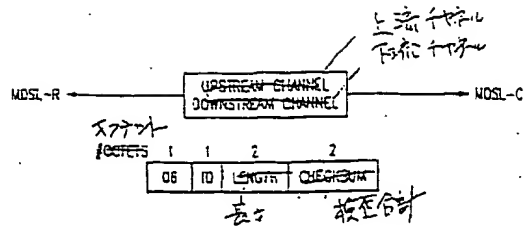
【図40】



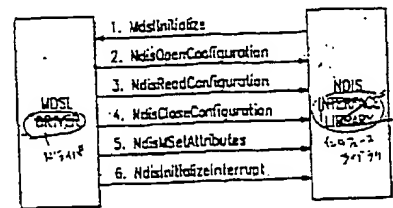
【図27】



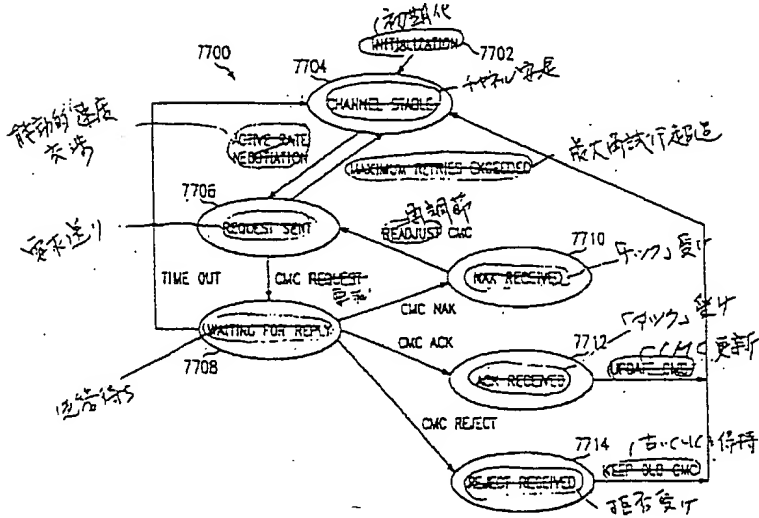
【図42】



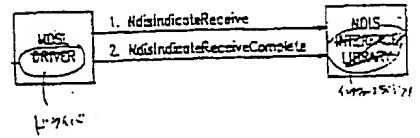
【図50】



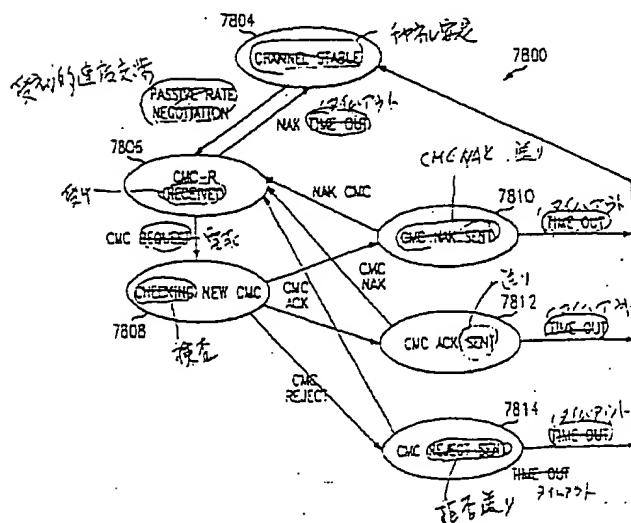
【図28】



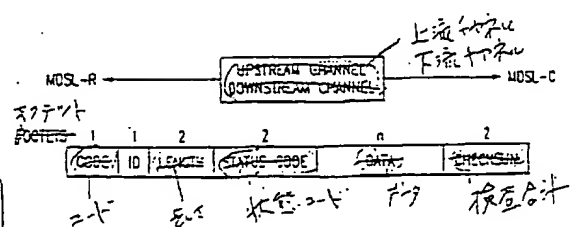
【図55】



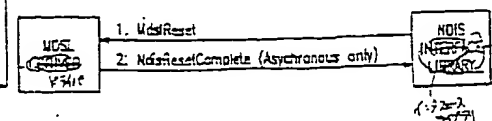
【図29】



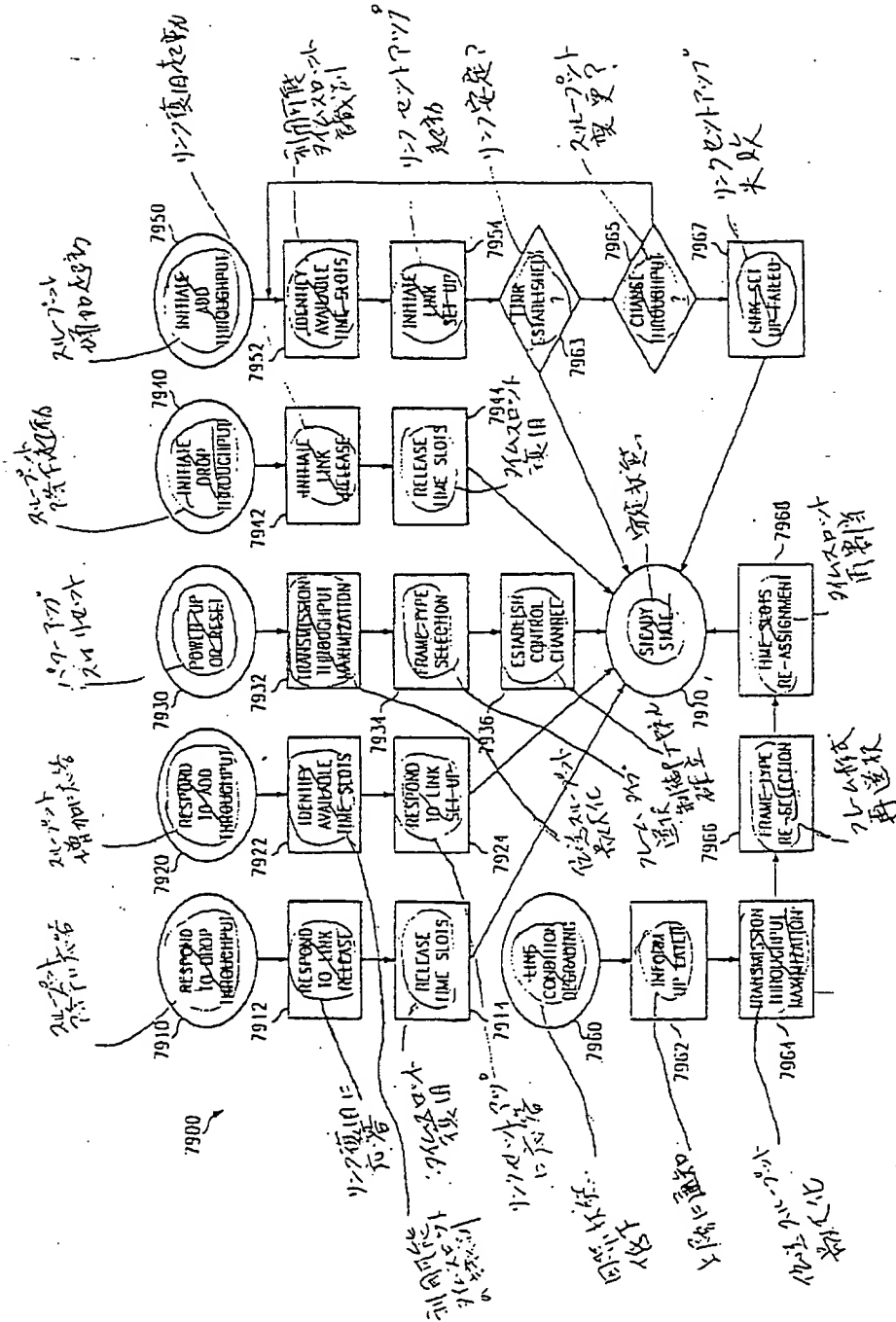
【図43】



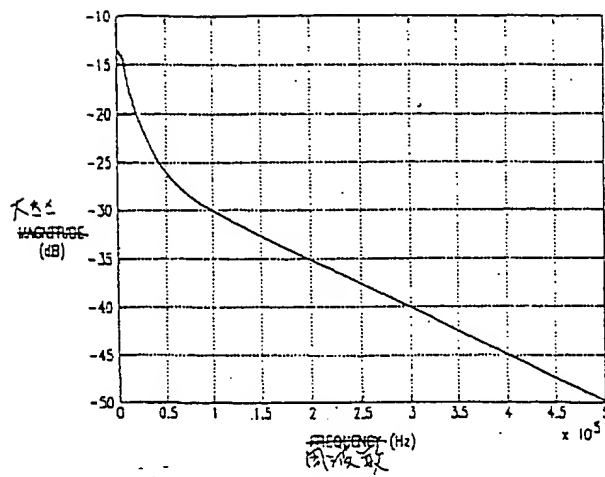
【図51】



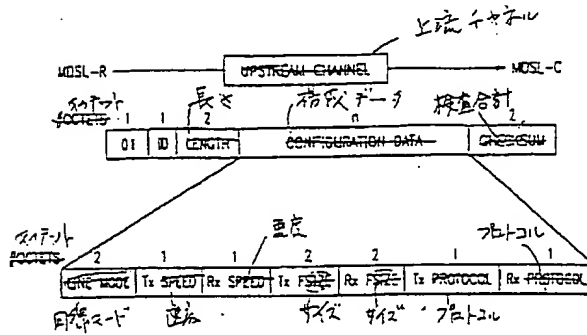
【図30】



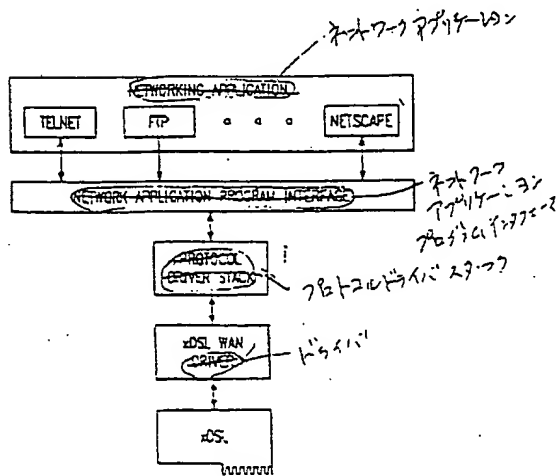
【図32】



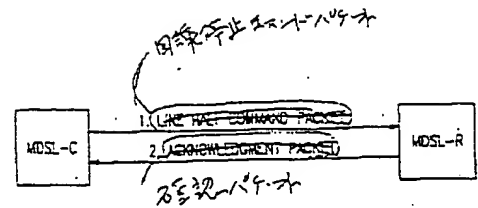
【図41】



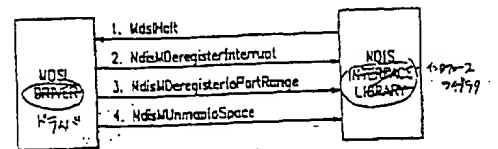
【図47】



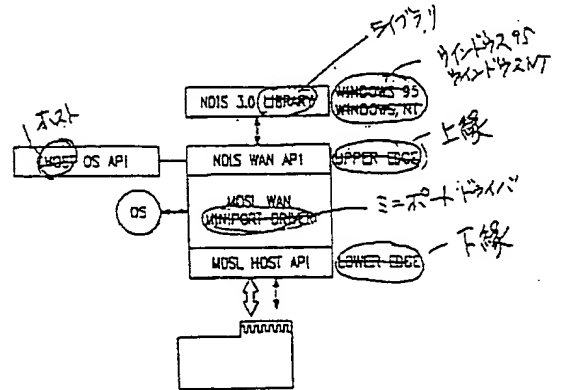
【図45】



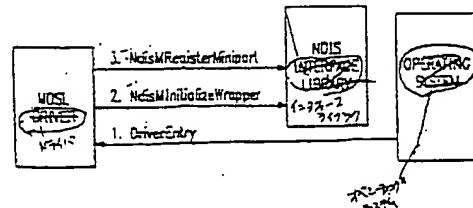
【図52】



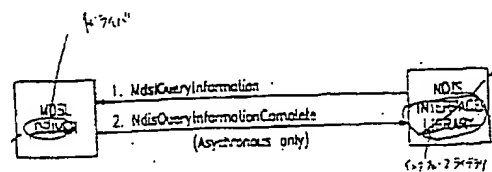
【図46】



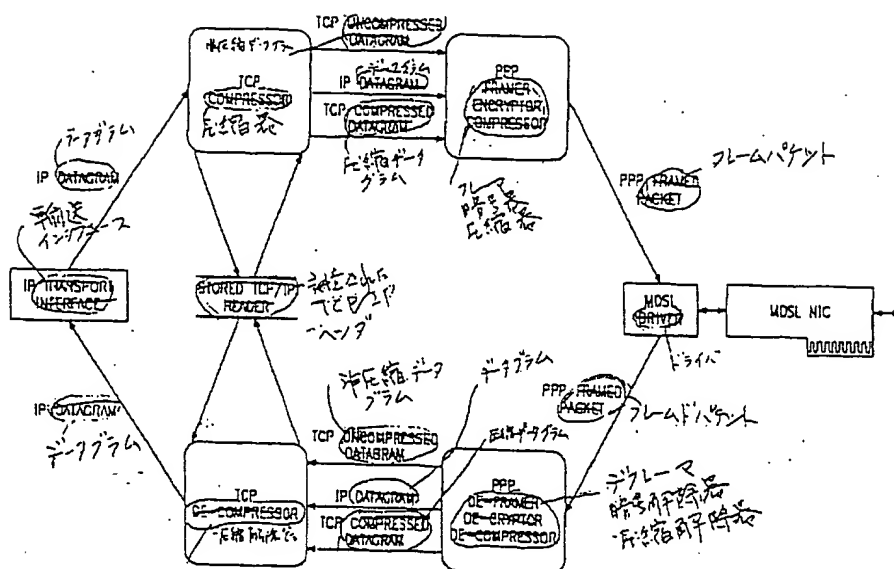
【図49】



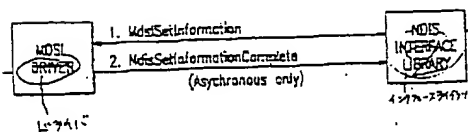
【図53】



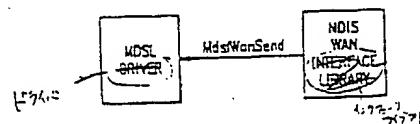
【图 48】



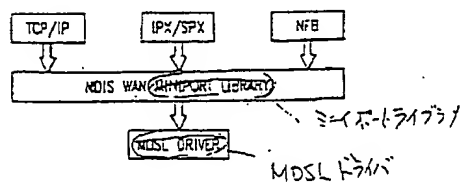
【図54】



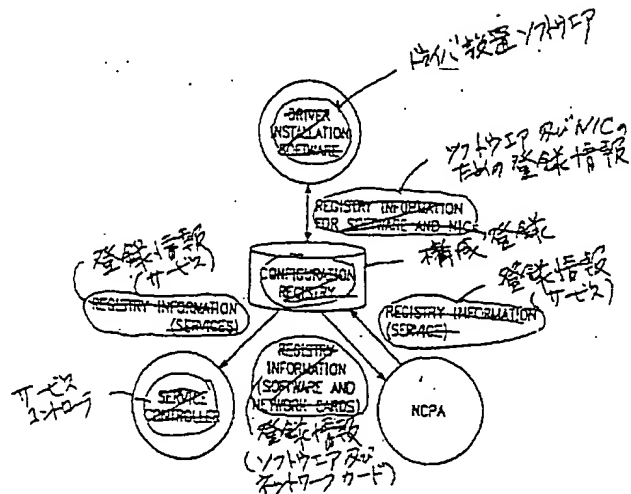
【図 5 6】



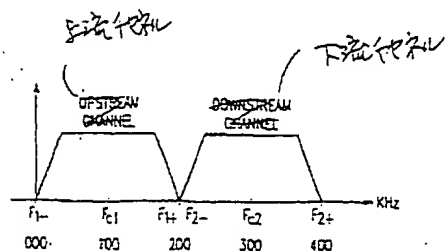
【圖57】



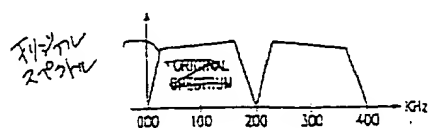
【图 58】



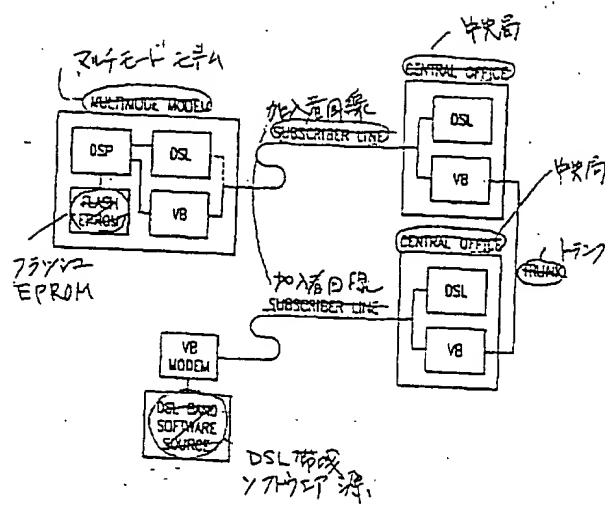
【图 60】



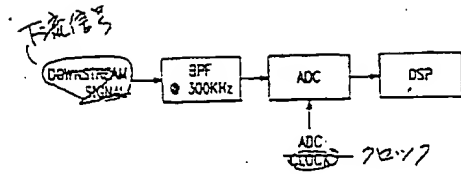
【図 6 4】



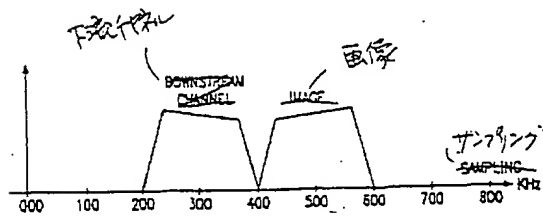
【図59】



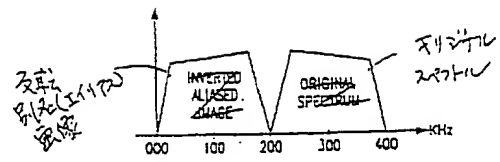
【図61】



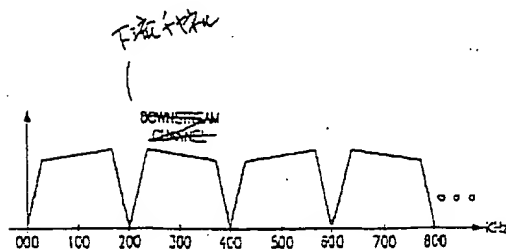
【図62】



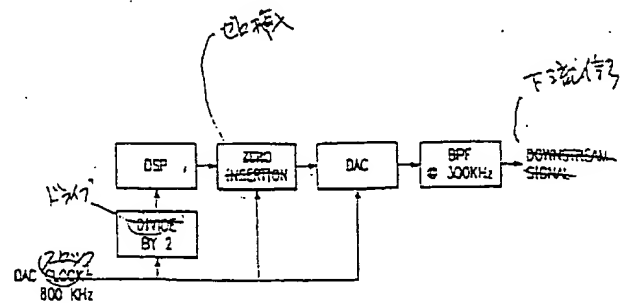
【図63】



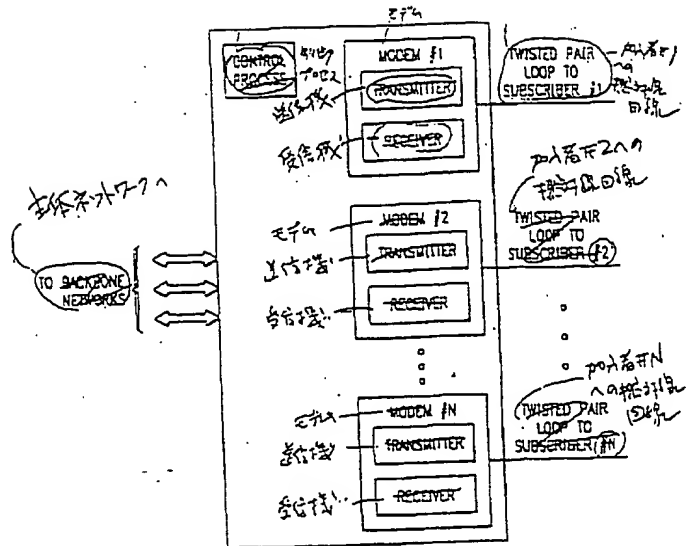
【図65】



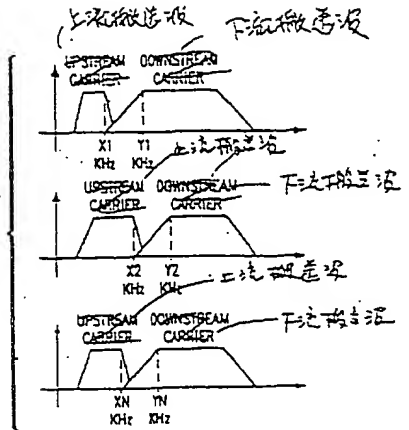
【図66】



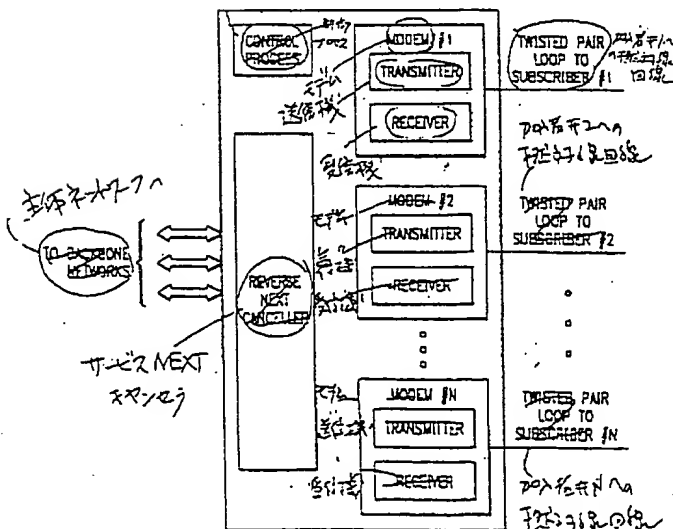
[図67]



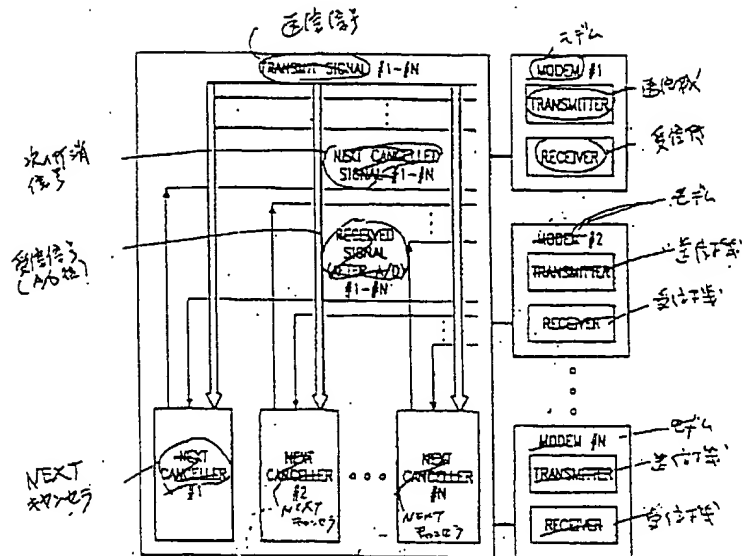
[図68]



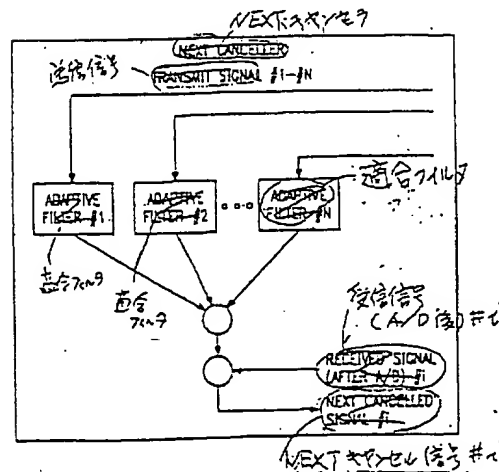
[図69]



【図70】



【図71】



フロントページの続き

- (72)発明者 ジーン エイ. フランツ
アメリカ合衆国テキサス州ミズーリ シティ,
ポイント クリアー コート 2027
- (72)発明者 ドミンゴ ジー. ガルシア
アメリカ合衆国テキサス州ブラノ, スナッ
ブドラゴン レーン 848
- (72)発明者 ザイアオリン ル
アメリカ合衆国テキサス州ダラス, フォレ
スト レーン 10010, ナンバー 531

- (72)発明者 デニス ジー. マナリング
アメリカ合衆国テキサス州ガーランド, バ
レイ クリーク ドライブ 1809
- (72)発明者 マイクル オー. ポーリイ
アメリカ合衆国テキサス州ガーランド, ル
ックアウト ドライブ 2609, ナンバー
7201

(72)発明者 テレンス ジェイ. リレイ
アメリカ合衆国テキサス州ロックウォー
ル, サンセット ヒル ドライブ 785

(72)発明者 ドナルド ビー. シェイバー
アメリカ合衆国テキサス州ダラス, フォー
ルメドウ レーン 7938

(72)発明者 ソング エス. ウ
アメリカ合衆国テキサス州ダラス, オウデ
リア ロード 12121, ナンバー 213